

INLG 2022

**International Natural Language Generation Conference
(INLG 2022)**

**Proceedings of the 15th International Conference on Natural
Language Generation**

July 17-22, 2022

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

Gold



Silver



Bronze



©2022 Association for Computational Linguistics

Order copies of this and other ACL proceedings from:

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

ISBN 978-1-955917-57-5

Preface by the General Chairs

We are delighted to present the Proceedings of the 15th International Natural Language Generation Conference (INLG 2022). After 2 years being held virtually due to the COVID-19 pandemic, this year the conference was a hybrid event, happening virtually and physically in Waterville, Maine, the United States between July 18-22, 2022.

INLG 2022 was locally organized by the Davis Institute for AI at Colby College thanks to the wonderful work of Amanda Stent, the Local Chair.

The INLG conference is the main international venue for the discussion of the computational task of Natural Language Generation (NLG) and its wide-range of applications, including data-to-text, text-to-text and vision-to-text approaches.

This year the conference consisted of a varied set of events. It started with a tutorial on "Artificial Text Detection", followed by the "NLG4Health" workshop whose proceedings included 4 accepted papers, a keynote speaker and a panel.

The main conference took place on July 19-21. Excluding Generation Challenges, we received a total of 51 committed submissions from which 19 were accepted as long papers, 6 as short papers and 4 as demo papers.

Generation Challenges, a set of shared tasks, was also presented during the main conference. The event proceedings consisted of 2 new shared-task proposals and the presentation of 3 completed ones. Besides the overview papers, the completed shared-tasks summed a total of 12 system descriptions.

This year INLG had 4 keynote speakers who did important contributions to the field:

- Dimitra Gkatzia, Edinburgh Napier University
- Emiel Krahmer, Tilburg University
- Margaret Mitchell, Huggingface
- Mohit Bansal, University of North Carolina (UNC) Chapel Hill

A panel on "Ethics and NLG" was also introduced in INLG 2022. We would like to thank the members, Nina da Hora, Sebastian Gehrmann, Sabelo Mhlambi, Nava Tintarev and Frank Schilder, as well as the moderator, Margaret Mitchell.

Last but not least, INLG 2022 closed with a hackathon on "Automatic Generation of Reports about the Gulf of Maine" on July 22nd.

The event was financially supported by:

- ARRIA (Gold)
- Google (Gold)
- HuggingFace (Silver)
- AX Semantics (Bronze)
- aiXplain (Bronze)

It is also important to mention that the 15th version of INLG would not be possible without the help of Area Chairs and Reviewers for whom we express our entire gratitude and that we relied on the expertise of Ehud Reiter and Emiel van Miltenburg, SIGGEN representatives.

Samira Shaikh
Thiago Castro Ferreira
INLG 2022 Programme Chairs

Organizing Committee

Program Chair

Samira Shaikh
Thiago Castro Ferreira

Local Chair

Amanda Stent

Invited Speakers

Dimitra Gkatzia, Edinburgh Napier University, UK
Emiel Kraemer, Tilburg University
Margaret Mitchell, HuggingFace
Mohit Bansal, University of North Carolina (UNC) Chapel Hill

SIGGEN Representatives

Ehud Reiter
Emiel Van Miltenburg

Tutorial and Hackathon Chair

Joshua Maynez

Publication Chair

Miruna Clinciu

GenChal Chair

Anastasia Shimorina

GenChal PC

Khyathi Chandu, CMU, US
Claire Gardent, CNRS/LORIA, France
Nikolai Ilinykh, University of Gothenburg, Sweden
Simon Mille, Universitat Pompeu Fabra, Spain
Shereen Oraby, Amazon, US

Sponsor Chair

Dave Howcroft

Social Media Chair

Luou (Lilly) Wen

Area Chairs

Albert Gatt
Chris van der Lee
Claire Gardent
Dimitra Gkatzia
Fei Liu
Malihe Alikhani
Michael White
Saad Mahamood
Tirthankar Ghosal
Yufang Hou

Local Organizing Team

Amy Poulin
Charlotte Buswick
Jake Rogers

Program Committee

Program Chairs

Thiago Castro Ferreira, Universidade Federal de Minas Gerais
Samira Shaikh, University of North Carolina, Charlotte

Area Chairs

Albert Gatt, Utrecht University
Chris Van Der Lee, Tilburg University and Tilburg University
Malihe Alikhani, University of Pittsburgh
Claire Gardent, CNRS
Dimitra Gkatzia, Edinburgh Napier University
Yufang Hou
Michael White, Ohio State University and Facebook
Fei Liu, University of Central Florida
Tirthankar Ghosal
Saad Mahamood, trivago N.V.

Keynote Talk: Advancing Natural Language Generation (for better or worse)

Margaret Mitchell
Huggingface

Abstract:

Bio: Margaret Mitchell is a researcher working on Ethical AI, currently focused on the ins and outs of ethics-informed AI development in tech. She has published over 50 papers on natural language generation, assistive technology, computer vision, and AI ethics, and holds multiple patents in the areas of conversation generation and sentiment classification. She currently works at Hugging Face driving forward work in the ML development ecosystem, ML data governance, AI evaluation, and AI ethics. She previously worked at Google AI as a Staff Research Scientist, where she founded and co-led Google's Ethical AI group, focused on foundational AI ethics research and operationalizing AI ethics Google-internally. Before joining Google, she was a researcher at Microsoft Research, focused on computer vision-to-language generation; and was a postdoc at Johns Hopkins, focused on Bayesian modeling and information extraction. She holds a PhD in Computer Science from the University of Aberdeen and a Master's in computational linguistics from the University of Washington. While earning her degrees, she also worked from 2005-2012 on machine learning, neurological disorders, and assistive technology at Oregon Health and Science University. She has spearheaded a number of workshops and initiatives at the intersections of diversity, inclusion, computer science, and ethics. Her work has received awards from Secretary of Defense Ash Carter and the American Foundation for the Blind, and has been implemented by multiple technology companies. She likes gardening, dogs, and cats.

Keynote Talk: NLG for Human-Robot Interaction: Challenges and Opportunities

Dimitra Gkatzia

Edinburgh Napier University

Abstract: Human-robot interaction (HRI) focuses on researching the interaction between humans and (mostly) physical robots. Despite the media coverage of robots displaying human-level capabilities in conversational dialogue and NLG, in reality, such robots use simple template-based approaches and follow pre-scripted interactions. In this talk, I will initially provide an overview of current approaches to NLG for HRI focusing on the limitations of current approaches and emphasising the challenges of developing flexible NLG approaches for HRI. Finally, I will provide an overview of our project CiViL: Commonsense and Visually-enhanced natural Language generation and discuss future directions.

Bio: Dimitra Gkatzia is an Associate Professor at the School of Computing at Edinburgh Napier University and a SICSA AI Theme co-lead. Dimitra is interested in making computers and robots interact in a human-like way using natural language, while at the same time respecting the privacy of the users. Her current work on Human-Robot Interaction focuses on the interplay between various modalities (vision, speech, knowledge-bases) in real-world settings for human-robot teaming scenarios. Her work in this area focuses on enhancing computers/robots' conversational capabilities with 'commonsense' similar to the ones present in human-human communication. She is also interested in NLG evaluation practices as well as addressing the challenges of NLG in low-resource settings where example relevant tools are scarce and data is not freely available, or it is hard to acquire due to challenges such as privacy issues or low numbers of native speakers of a language.

Keynote Talk: Modeling and Evaluating Faithful Generation in Language and Vision

Mohit Bansal

University of North Carolina (UNC) Chapel Hill

Abstract: Faithfulness is a key aspect of accurate and trustworthy generation in diverse modalities such as language and vision. In this talk, I will present work towards modeling and evaluating faithfulness in summarization and multimodal tasks. First, we will discuss our earlier work on multi-task and reinforcement learning methods to incorporate auxiliary faithfulness-promoting skills such as entailment and back-translation validity. We will then describe abstractive summarization models that holistically address the problem of faithfulness during pre-training and fine-tuning. Next, we will explore improved summary faithfulness evaluation methods based on human-automation balance and semantic graph representations. Lastly, we will briefly discuss faithful, fine-grained skill evaluation of text-to-image generation models.

Bio: Dr. Mohit Bansal is the John R. Louise S. Parker Professor in the Computer Science department at University of North Carolina (UNC) Chapel Hill. He received his PhD from UC Berkeley and his BTech from IIT Kanpur. His research expertise is in natural language processing and multimodal machine learning, with a particular focus on grounded and embodied semantics, human-like language generation and QA/dialogue, and interpretable and generalizable deep learning. He is a recipient of the DARPA Director's Fellowship, NSF CAREER Award, and Army Young Investigator Award. His service includes ACL Executive Committee, ACM Doctoral Dissertation Award Committee, Program Co-Chair for CoNLL 2019, ACL Americas Sponsorship Co-Chair, and Associate/Action Editor for TACL, CL, IEEE/ACM TASLP, and CSL journals.

Keynote Talk: From Generating Personalised Risk Descriptions to Data-driven Health Narratives: Applying NLG to Healthcare Data

Emiel Krahmer
Tilburg University

Abstract: Even though the health domain has been mentioned as a potential application domain for NLG since the early days of the field, one thing that has changed recently is the emergence of large and growing amounts of patient-generated health data in hospitals and other care facilities (think of self-reported outcome measures, electronic health records, registry data). Hospitals are urgently looking for ways to make this data accessible in a personalised manner for both patients and clinicians. Naturally, this is a task for which NLG lends itself very well. In this talk, I describe how traditional NLG tasks, such as the generation of referring expressions and of narratives, re-emerge in this health domain. I also highlight which particular evaluation questions this raises, such as, for example, what kind of information do patients actually want? In which format do they prefer this information? What cognitive implications does access to this information have? In the final part of the talk I touch upon a number of broader questions, such as is there room for transformers in this application domain, and what are ethical and privacy issues for this kind of application.

Bio: Prof. Dr. Emiel Krahmer is a full professor in the Tilburg School of Humanities and Digital Sciences. In his research, he studies how people communicate with each other, and how computers can be taught to communicate in a similar fashion, to improve communication between humans and machines. His current research is positioned at the intersection of artificial intelligence, natural language processing and human communication studies, and has applications in, for example, media (e.g., automatic moderation and summarization of online discussions), health (e.g., data-driven treatment decision aids; chatbots for smoking cessation) and education (e.g., social robots teaching children a second language).

Table of Contents

<i>Evaluating Referring Form Selection Models in Partially-Known Environments</i> Zhao Han, Polina Rygina and Thomas Williams	1
<i>Template-based Approach to Zero-shot Intent Recognition</i> Dmitry Lamanov, Pavel Burnyshev, Katya Artemova, Valentin Malykh, Andrey Bout and Irina Piontkovskaya	15
<i>"Slow Service" → "Great Food": Enhancing Content Preservation in Unsupervised Text Style Transfer</i> Wanzheng Zhu and Suma Bhat	29
<i>Arabic Image Captioning using Pre-training of Deep Bidirectional Transformers</i> Jonathan Emami, Pierre Nugues, Ashraf Elnagar and Imad Afyouni	40
<i>Plot Writing From Pre-Trained Language Models</i> Yiping Jin, Vishakha Kadam and Dittaya Wanvarie	52
<i>Paraphrasing via Ranking Many Candidates</i> Joosung Lee	68
<i>Evaluating Legal Accuracy of Neural Generators on the Generation of Criminal Court Dockets Description</i> Nicolas Garneau, Eve Gaumont, Luc Lamontagne and Pierre-Luc Déziel	73
<i>Automatic Generation of Factual News Headlines in Finnish</i> max.koppatz@gmail.com max.koppatz@gmail.com, Khalid Alnajjar, Mika Hämäläinen and Thierry Poibeau	100
<i>Generating Coherent and Informative Descriptions for Groups of Visual Objects and Categories: A Simple Decoding Approach</i> Nazia Attari, David Schlangen, Martin Heckmann, heiko.wersing@honda-ri.de heiko.wersing@honda-ri.de and Sina Zarrieß	110
<i>Dealing with hallucination and omission in neural Natural Language Generation: A use case on meteorology.</i> Javier González Corbelle, Alberto Bugarín-Diz, Jose Maria Alonso-Moral and Juan Taboada	121
<i>Amortized Noisy Channel Neural Machine Translation</i> Richard Yuanzhe Pang, He He and Kyunghyun Cho	131
<i>Math Word Problem Generation with Multilingual Language Models</i> Kashyapa Niyarepola, Dineth Athapaththu, Savindu Kalsara Ekanayake and Surangika Ranathunga	144
<i>Comparing informativeness of an NLG chatbot vs graphical app in diet-information domain</i> Simone Balloccu and Ehud Reiter	156
<i>Generation of Student Questions for Inquiry-based Learning</i> Kevin Ros, Maxwell Jong, Chak Ho Chan and ChengXiang Zhai	186
<i>Keyword Provision Question Generation for Facilitating Educational Reading Comprehension Preparation</i> Ying-Hong Chan, Ho-Lam Chung and Yao-Chung Fan	196

<i>Generating Landmark-based Manipulation Instructions from Image Pairs</i> Sina Zarrieß, Henrik Voigt, David Schlangen and Philipp Sadler	203
<i>Zero-shot Cross-Linguistic Learning of Event Semantics</i> Malihe Alikhani, Thomas H Kober, Bashar Alhafni, Yue Chen, Mert Inan, Elizabeth Kaye Nielsen, Shahab Raji, Mark Steedman and Matthew Stone	212
<i>Nominal Metaphor Generation with Multitask Learning</i> Yucheng LI, Chenghua Lin and Frank Guerin	225
<i>Look and Answer the Question: On the Role of Vision in Embodied Question Answering</i> Nikolai Ilinykh, Yasmeen Emampoor and Simon Dobnik	236
<i>Strategies for framing argumentative conclusion generation</i> Philipp Heinisch, Anette Frank, Juri Opitz and Philipp Cimiano	246
<i>LAFT: Cross-lingual Transfer for Text Generation by Language-Agnostic Finetuning</i> Xianze Wu, Zaixiang Zheng, Hao Zhou and Yong Yu	260
<i>Quantum Natural Language Generation on Near-Term Devices</i> Amin Karamlou, James R Wootton and Marcel Pfaffhauser	267
<i>Towards Evaluation of Multi-party Dialogue Systems</i> Khyati Mahajan, Sashank Santhanam and Samira Shaikh	278
<i>Are Current Decoding Strategies Capable of Facing the Challenges of Visual Dialogue?</i> Amit Kumar Chaudhary, Alex J. Lucassen, Ioanna Tsani and Alberto Testoni	288
<i>Analogy Generation by Prompting Large Language Models: A Case Study of InstructGPT</i> Bhavya Bhavya, Jinjun Xiong and ChengXiang Zhai	298

Evaluating Referring Form Selection Models in Partially-Known Environments

Zhao Han and Polina Rygina and Tom Williams

MIRRORLab

Department of Computer Science

Colorado School of Mines

zhaohan@mines.edu, prygina@mines.edu, twilliams@mines.edu

Abstract

For autonomous agents such as robots to effectively communicate with humans, they must be able to refer to different entities in situated contexts. In service of this goal, researchers have recently attempted to model the selection of *referring forms* on the basis of cognitive status (informed by Givenness Hierarchy), and have shown promising results with over 80% accuracy. However, we argue that the task environments lack ecological validity, due to their use of a small number of objects that are constantly activated and easily uniquely identifiable. Accordingly, we present a novel building-construction task that we believe has increased ecological validity. We then show how training cognitive status informed referring form selection models on data collected within this novel task environment yields substantially different results from those found in previous work, providing key insights and directions for future work.

1 Introduction

One of the most studied dimensions of natural language pragmatics is *reference*: the process by which speakers pick out, or *refer*, to things of interest in the environment, and how hearers interpret, or *resolve* those references. The generation (or production) side of this problem has attracted sustained attention across a variety of communities, including philosophy of language, psycholinguistics, and artificial intelligence – so much so that referring has been called the “fruit fly” of language (Van Deemter, 2016).

The vast majority of research on referring, however, has been focused on problems like *Referring Expression Generation* (Krahmer and Van Deemter, 2012), in which the goal is to select the *properties* that will be used in a generated expression (e.g., choosing to highlight the redness, or the boxiness, of a red box, among other possible properties). In contrast, very little research has been done on the

problem of computationally modeling *Referring Form Selection*, in which a speaker must select a more general *referring form*, such as “it”, “that”, or “the $\langle N' \rangle$ ”¹, despite its accepted status as an important initial step during language production (Kibrik, 2011).

While the computational modeling of referring expression generation has been heavily understudied, it has been a key question of interest in the linguistics community, with a number of competing theories making different predictions, including Accessibility Theory (Ariel, 2001) and Givenness Hierarchy Theory (Gundel et al., 1993). Such theories thus provide natural starting points for computational modeling work. Yet while these theories provide critical linguistic insights about the nature of referring form selection, they provide little direct input into the cognitive processes, mechanisms, or algorithms that govern this process.

Recently, this has begun to change, with researchers like Pal et al. (2020) seeking to directly computationally model the mechanics of these underlying theories of reference (in their case, *Givenness* or *Cognitive Status*), and then build higher-level computational models of referring form selection that leverage those more fundamental models (Pal et al., 2021). These recent works have provided promising results, with over 80% accuracy in predicting the referring forms used by interactants in human-human and human-robot interaction scenarios.

Yet despite the promise of these results, concerns may be raised about the task environments in which those results were produced. Specifically, we argue that the task environment used in that previous work was not ideally suited for training or evalu-

¹In this work we implicitly focus on Standard American English; but the types and distribution of general referring forms we consider have been observed across a wide variety of languages beyond English, including Mandarin, Japanese, Spanish, Russian, Eegimaa, Kумык, Ojibwe, Arabic, Irish, Norwegian, Persian, and Turkish (Hedberg, 2013).

ating Cognitive Status informed Referring Form Selection models.

In this paper, we thus measure the performance of these previously published models using a better collection of tasks, making three key contributions in the process: (1) we present a novel task context that we argue is well designed for the studying of referring form selection; (2) we assess the performance of Pal et al. (2021)’s Referring Form Selection model in this setting to obtain a better estimate of its true performance in realistic task contexts; and (3) we use these results to motivate arguments as to how underlying models of cognitive status must be adapted to enable better performance on Referring Form Selection tasks.

2 Related Work

We will now describe prior what work has been done on Referring Form Selection, including the Cognitive Status informed work of Pal et al. (2021). We will then provide our specific critiques of the task context in which that work was trained and evaluated.

Referring Form Selection models fall into two main categories (Arnold and Zerkle, 2019). *Rational* models seek to explain how speakers egocentrically decide whether or not to use pronouns, e.g. for reasons of ease of production (Aylett and Turk, 2004; Frank and Goodman, 2012). In contrast, *pragmatic* models seek to explain how speakers allocentrically decide to use pronouns on the basis of their status as activated or focused within a conversation (Grosz et al., 1995; Brennan et al., 1987; Ariel, 1991; Gundel et al., 1993). These pragmatic models share an assumption that referring form selection is grounded in a relationship between discourse context and mnemonic or attentional states. For example, Gundel et al. (1993) suggest that referring forms are selected based on which of a hierarchically nested set of *Cognitive Statuses* ($\{\text{in focus} \subseteq \text{activated} \subseteq \text{familiar} \subseteq \text{uniquely identifiable} \subseteq \text{referential} \subseteq \text{type identifiable}\}$) can be assumed to hold for the target referent.

While these models have shown promise in predicting whether or not someone chooses to use a definite noun phrase or a more reduced form, neither class of model is terribly effective at predicting precisely which form a speaker will choose to use. Rational models, for example, predict much more frequent use of reduced forms than are actually seen in practice, and fail to predict differential use

of “equally short” referring forms (Arnold and Zerkle, 2019). To make matters work, models in both categories tend to focus on specific referential phenomena, rather than trying to comprehensively model the entire process of reference production (Arnold and Zerkle, 2019; Grüning and Kibrik, 2005); and indeed often do not really try to model cognitive mechanisms or psycholinguistic processes at all (Arnold, 2016). And, of critical importance to those studying *situated* interaction, the vast majority of this previous work, in both camps, has predominantly been assessed on corpora not collected in or encoding any features of situated domains.

Work in the Artificial Intelligence community on Referring Form Selection suffers from similar problems. Most such work (Poesio et al., 2004; McCoy and Strube, 1999; Ge et al., 1998; Kibrik et al., 2016; Kibrik, 2011; Callaway and Lester, 2002; Kibble and Power, 2004) falls under *multi-factorial process modeling*, in which the process of referring is modeled as a classification problem performed on the basis of a variety of features (Kibrik, 2011; Van Deemter et al., 2012; Gatt et al., 2014). Like the linguistic models discussed above, these models often do not attempt to select between referring forms at a fine-grained level, instead choosing to predict pronoun use as a whole. And, like the linguistic models discussed above, these models are often trained and evaluated in purely textual domains, such as the Wall Street Journal corpus (Krasavina and Chiarcos, 2007), thus avoiding many of the nuanced challenges that arise in situated domains, which are highly ambiguous and open worlds, and in which agents must make decisions on the basis of features that can be readily and immediately assessed, which may well go beyond purely linguistic features, including features of the environment in which dialogue is situated.

Some recent research efforts have attempted to fix these problems. Pal et al. (2020), for example, have presented models for dynamic modeling of Cognitive Status (a construct underlying Givenness Hierarchy theoretic accounts of referring (Gundel et al., 1993)), and have then used these models as informative features for Referring Form Selection, with apparently good results (Pal et al., 2021). Pal et al.’s work is also notable in that it is trained on data collected in situated interaction contexts. However, even this work suffers from certain flaws that may raise similar questions about generalizability

to situated domains. Specifically, we argue that Pal et al.’s work was conducted in a task environment that may not have been well suited for training or evaluation of cognitive status informed models. Pal et al.’s model was trained and evaluated using videos collected by Bennett et al. (2017), in which humans give instructions to humans or robot interactants as to how to re-arrange a large-scale environment to match a pre-determined pattern.

This task domain may be ill-suited to studying Cognitive Status informed Referring Form Selection for several reasons. First, this domain contains a relatively small number of candidate referents, i.e., three towers of cans and four labeled boxes. This could result in an irregular situation in which the majority of task-relevant objects are constantly at least *activated* (which, in a Givenness Hierarchy theoretic account, would enable the use of referring forms such as *this*), and are likely to remain so regardless of dialogue context merely due to the small number of observed task relevant objects. Second, all task-relevant objects in this domain are easily uniquely discriminable. Each of the “towers” has a unique context, and each of the boxes is labeled with a unique letter. This means that speakers may be able to over-rely on proper nouns and simple single-property descriptions, and would not need to seriously consider their choice of referring expression. Third, all task-relevant objects in this domain are visible at all times. This is likely to exacerbate the challenges listed above. Moreover, it is likely to completely preclude the need for indefinite descriptions, which are often used when the speaker assumes that the listener does not already have knowledge of their target referent.

In this work, we seek to address these challenges. We begin by collecting a new corpus of sequential referring expressions in a task context that does not have these shortcomings. Then, we re-assess the performance of Pal et al. (2021)’s Referring Form Selection models on the data collected in this more ideally suited task domain.

3 Environment and Task Design

To collect a wider variety of referring forms in a situated context, we designed a dyadic interaction task (Shown in Figure 1) in which pairs of participants perform four tower construction tasks in four visually separated quadrants of a larger task environment. The task environment is separated into four quadrants to create a partially-observable envi-

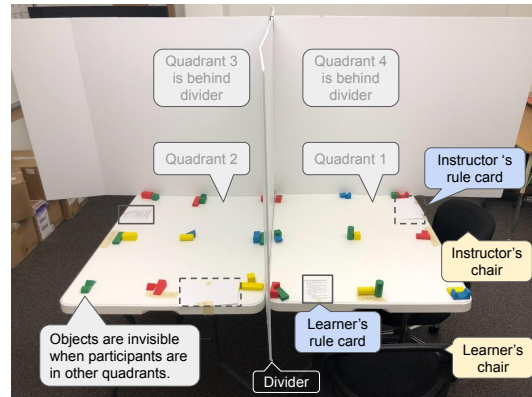


Figure 1: Two of four quadrants of the task environment. To promote a wide variety of referring forms, we placed objects in different quadrants with careful manipulation of target referent visibility (thus leading to course-grained variance in cognitive status) and by requiring repeated reference to task referents (thus leading to fine-grained variance in cognitive status).

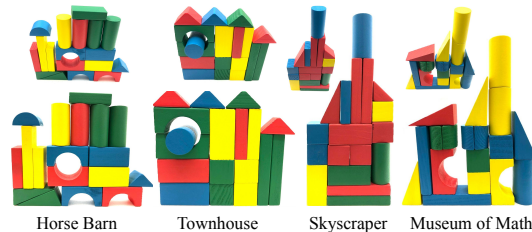


Figure 2: The buildings in the construction task. Two angles were provided to help participants to recognize constituent block shapes.

ronment in which participants can readily observe their current quadrant, but not the other quadrants. Each quadrant is filled with block shapes, including triangles, cubes, cuboids, cylinders, arches, and half-circles. All blocks are distributed to the corners and intersections of a 3×3 grid.

The described task environment is used as the setting for a series of four dyadic construction tasks, one in each of the four quadrants. Each task requires one participant (the Teacher) to instruct the other participant (the Learner) to construct a building based on a given image (Figure 2). The Learner, in turn, must work to construct the tower piece by piece as it is described to them, without speaking themselves, using only the resources available in their current quadrant unless the Teacher instructs them to seek a block in a different quadrant. Note that participants do not statically provide or listened to monolithic multi-minute monologues. In fact, the task is highly interactive, with teachers giving instructions, learners following instructions, and

then teachers providing corrections or proceeding. While learners were mostly silent while completing their tasks, this is perfectly reasonable given the particular domain we investigate in this work, i.e., deciding how to deliver multi-step task instructions.

Each building has 18 blocks, nine (50%) of which are placed in the quadrant where the building is being constructed, the other half of which are distributed in the other three quadrants. The large number of blocks in this task context ensure that, unlike in Pal et al.’s work, there are a large number of candidate referents that are not trivially distinguishable. The separation between quadrants ensures that, unlike in Pal et al.’s work, not all objects are visible at any given time. And, the distribution of blocks throughout the four quadrants ensures that the Teacher will need to refer to blocks that have not yet have been observed or which were observed in a previous construction task but which are no longer visible in the current quadrant, further diversifying the expected set of referring expressions used by Teachers.

4 Corpus Collection Procedure

The described environmental and task context were used to collect a new corpus of referring expressions, through the following IRB-approved procedure. Eleven pairs of participants were recruited from the campus of The Colorado School of Mines. Upon arrival, each pair of participants provided informed consent and were provided instructions about the structure of the tower construction task. Participants were then led to the task environment and seated in the first quadrant, where a photo of the target building was available to the participant assigned to be the Teacher, as seen in Figure 1. Participants were then videorecorded completing each of the four tower construction tasks in sequence. Each participant was paid \$10 USD.

5 Corpus Annotation

The collected eleven-dyad corpus was comprised of eleven collections of four monologues each. These eleven collections averaged 27:32 minutes in length, with a minimum of 16:26 and maximum of 34:03. The average monologue length was 6:53. We first transcribed these recordings automatically online using the Dovetail qualitative analysis software². The first two authors then manually veri-

²<https://dovetailapp.com/>

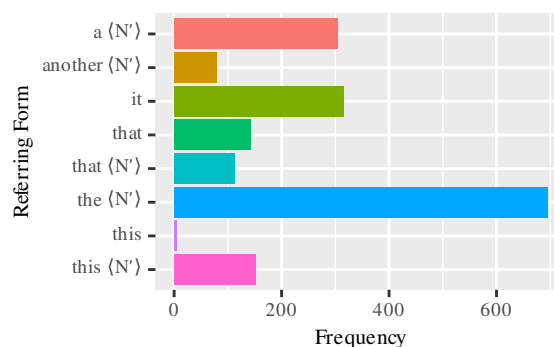


Figure 3: Distribution of a wide variety of referring forms. In addition to the six definitive nouns (right six columns), we also found participant frequently used indefinite nouns of “a ⟨N'⟩” and “another ⟨N'⟩” (left two columns).

fied and corrected these transcripts. The collected transcripts were then divided into a total of 1992 utterance clauses.

After removing clauses that contained no referring forms (e.g., utterances made when switching quadrants and at study conclusion) or only plural referring forms (e.g., them, they), which we did not aim to model in this work, 1867 referring expressions remained, including from the corrective instructions. Each participant contributed an average of 169.7 referring forms, which is significantly more than the average of 18 (603/33) referring forms per participant in the situated interaction corpus (Bennett et al., 2017) used by Pal et al. (2021). The data does not have information that names or uniquely identifies individual people or offensive content. Below, we provide two sample utterance sequences from the collected corpus.

Sample 1

- *Alright. Do you see the red block over there?*
- *We need that but the blue.*
- *Awesome. Put that leg on that side to the left of the green cube, just like that.*
- *And you see the red thing I was talking about?*
- *Put that right on top of the blue thing. Perfect.*

Sample 2

- *And then the blue cylinder is going to go there.*
- *Okay. So for the next, we need this one.*
- *And you can go ahead and set that up right next to the triangle.*
- *And put it vertically on the inside.*

After translating each of the corpus’ 44 monologues into a sequence of (non-plural) referring

forms, we categorized each into one of eight categories (See Figure 3), and annotated, at each reference point, key features of each candidate object in the environment. Critically, we ensured that the features used could all be assessed on the fly, to ensure they could actually be used in future robotics applications. In the following subsections, we detail both of these types of annotation.

5.1 Referring Forms

We categorized referring expressions into seven types of referring forms: *it*, *this*, *that*, *this* $\langle N' \rangle$, *that* $\langle N' \rangle$, *the* $\langle N' \rangle$ and \langle *indefinite NP* \rangle . While indefinite noun phrases took multiple forms (e.g., “a $\langle N' \rangle$ ”, which accounted for 16.3% of all referring forms, and “another $\langle N' \rangle$ ”, which accounted for 4.2% of referring forms (per Figure 3). Similarly, like (Pal et al., 2020), we take a descriptivist view (Frege, 1892; Russell, 2001; Nelson, 2002) and merge bare noun phrases together with definite noun phrases.

5.2 Object Features

Next, we discuss the features annotated for each object in the scene at each reference point. We used the same four simple features used with great success by Pal et al. (2021), both because they are easily assessable by autonomous agents like robots, and to facilitate direct comparison with Pal et al. (2021). Each of these four features is described in a subsection below.

5.2.1 Cognitive Status

The first feature used was Cognitive Status, which was, unsurprisingly, the most informative feature used in Pal et al. (2021)’s Cognitive Status informed approach. To annotate the cognitive status of each object in the scene at each point of reference, we used the Cognitive Status model used by Pal et al. (2021), as defined in Pal et al. (2020). This approach uses a *Cognitive Status Engine* comprised of a set of *Cognitive Status Filters*, one for each object. Each Cognitive Status Filter is a Bayesian filter of the form:

$$p(S_o^t) = p(S_o^{t-1})p(L_o^t)p(S_o^t | S_o^{t-1}, L_o^t)$$

Here, S is a cognitive status in $\{I, A, F\}$ (where I is “In Focus”, A is “Activated”, and F is “Familiar”, predicted from an object’s cognitive status at the previous time point and the object’s “linguistic status” at the previous timepoint $L \in \{N, M, T\}$ (where N is “Not Mentioned”, M is “Mentioned”, and T is “Mentioned in a Topic Role”. To compare

FL: 6	FM: 5	FR: 6
ML: 4	MM: 3	MR: 4
NL: 2	NM: 1	NR: 2
	Instructor	

Table 1: Codes for physical distance.

directly with Pal et al. (2020) and Pal et al. (2021), we have made the same assumption that all objects are initially at least familiar. This is a simplifying assumption that we will return to later.

Initially, using this model identically to how it was used by Pal et al. (2021) failed to predict any objects in the scene to be “In Focus” at any timepoint. To diagnose this problem, we created a *blended model* by linearly combining a non-probabilistic model HL directly derived from linguistic rules (cp. (Pal et al., 2021)) with the probabilistic model C trained by Pal et al. (2021): $C' = w_1HL + w_2C$, where $w_l = 0.1$, $w_c = 0.9$. Here, HL is encoded as a 9×3 matrix ($S \times L$) where each row (column, as transposed below) represented a combination of a cognitive status and linguistic status at time $t - 1$, and each column (row, as transposed below) represented a cognitive status at time t :

$$L^T = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

This *blended model* was then used as the basis for each Cognitive Status Filter.

5.2.2 Number of Distractors

Next, we considered the number of distractors, which are the number of objects that have a cognitive status at the same GH-theoretic tier or higher than the target. We believe the number of distractors affects how people determine referential choice, as evidenced by Ferreira et al. (2005).

5.2.3 Physical distance

During our task design phase, we have intentionally placed blocks at a 3×3 grid, allowing us to classify each referring form in at least nine categories, a combination of {near (N), middle (M), far (F)} and {left (L), middle (M), right (R)}. The nine grid points are coded as shows in Table 1 (note that a participant sits below NM).

Additionally, each object can be in one of four distance-relevant task categories at any time: on-table (T), in-building (B), in-hand (H), and in-other-quadrant (O). Although B and H are specific to our

Model	Removed Feature
M1	N/A (full model)
M2	Cognitive status
M3	Number of distractors
M4	Physical distance
M5	Temporal distance

Table 2: Five model types.

task scenario, they can be generalized: B can be seen as objects at the *task goal location*, and H can be generalized to *invisible locations*. Because T is a general term, we coded it the same as MM, i.e., 3. B and H do not have distance comparisons, we coded them as 0. As O is furthest, we coded it as 10. This was a simplifying assumption to best compare with prior work.

5.2.4 Temporal Distance

Similar to Pal et al. (2021), we annotated recency of mention, i.e., temporal distance, for each object by indexing the previous occurrence of the object. TD is coded as 0 when an object is not yet mentioned in a monolog, 1 when the object is the last mentioned object, and $1/n$ where n is the number of objects referred since the object was mentioned.

6 Computational Modeling

As we intended to interrogate the performance of previous published models, we use the same decision tree algorithm by Pal et al. (2021) for explainability and theory-building purposes. Specifically, we used the same decision tree implementation in Weka 3.8.6 (Eibe et al., 2016): REPTree (Reduced Error Pruning Tree) (Quinlan, 1987), an extension of the C4.5 algorithm. REPTree builds a decision tree using information gain and prunes the tree using reduced-error pruning (REP) with backfitting (Witten and Frank, 2002).

Similarly, we followed the same training procedure as Pal et al. (2021), training five distinct models (Table 2): a full model (M1), and four ablated models, removing either cognitive status (M2), distractors (M3), physical distance (M4), or temporal distance (M5). We initially set the maximum depth of the tree to six, the same as Pal et al.’s model, but the decision tree became complex and difficult to interpret/explain, we thus set the maximum allowed depth of the tree to five. Similar performance was observed at depth 5 vs. 6.

The performance of these five models (for unpruned and pruned trees) were evaluated using five-fold cross validation to further avoid over-fitting

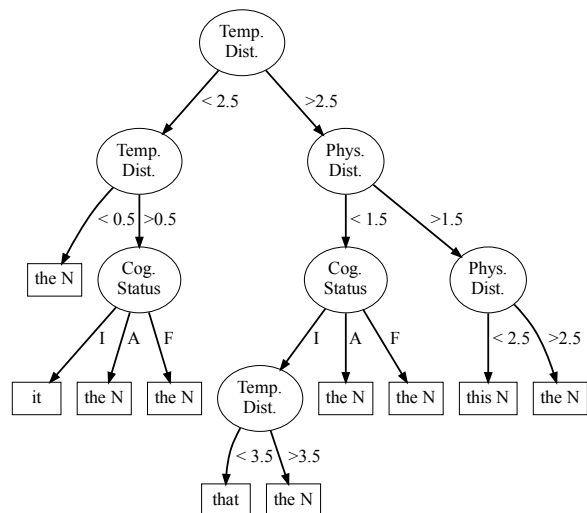


Figure 4: The decision tree visualization for the best-performing M1 (six main referring form categories informed by Givenness Hierarchy). {I,A,F}={In Focus, Activated, Familiar}.

(note that the tree pruning is also for this purpose). To quantify the models’ performance, five common scoring metrics are used, as in (Pal et al., 2021): accuracy, root mean squared error (RMSE), precision, recall, and F1 score. The latter three are weighted by class size. Additionally, we used coverage (modeled as number of classes included in model predictions) and number of leaves to quantify model simplicity and explainability.

All data (which will be licensed under CC-BY 4.0) and code are attached to this submission.

6.1 Results

Table 3 shows the results for the full, unpruned trees; Table 4 shows the results for the pruned trees, which had lower coverage but are more readily interpretable. In both tables, the left and right sides show results with and without indefinite forms included. We consider these separately as Pal et al. (2021) did not consider indefinite noun phrases.

In this section we will more deeply interrogate the results of the pruned trees, as they are more readily interpretable. As seen in Table 4 left, we achieved 61%–66% accuracy for M1-M5. M1 and M3 (removing the number of distractors) are top-performing on all metrics. For M5 where the temporal distance feature was not used, the performance is slightly dropped to 61.72%. All models scored similarly in other metrics.

Fig. 4 shows a tree visualization for the M1 model. From the top node, it branches at the tem-

	Six GH informed referring forms					With two indefinite forms				
	M1	M2	M3	M4	M5	M1'	M2'	M3'	M4'	M5'
Accuracy	66.01	63.41	65.87	61.58	61.08	59.50	59.00	59.67	51.02	57.17
RMSE	0.340	0.366	0.341	0.384	0.389	0.405	0.410	0.403	0.490	0.428
Precision	0.573	0.527	0.572	0.514	0.542	0.509	0.506	0.512	0.432	0.498
Recall	0.660	0.634	0.659	0.616	0.611	0.595	0.590	0.597	0.510	0.572
F1 score	0.597	0.571	0.596	0.546	0.560	0.544	0.539	0.545	0.454	0.522
Coverage	5	4	4	5	5	6	6	6	6	6
Leaves	35	31	34	29	16	35	31	30	30	23

Table 3: Evaluation metrics and results for unpruned trees.

	Six GH informed referring forms					With two indefinite forms				
	M1	M2	M3	M4	M5	M1'	M2'	M3'	M4'	M5'
Accuracy	65.73	64.11	65.80	62.98	61.72	59.83	58.95	59.83	51.30	57.29
RMSE	0.343	0.359	0.342	0.370	0.383	0.402	0.411	0.402	0.487	0.427
Precision	0.552	0.543	0.552	0.509	0.521	0.493	0.487	0.493	0.435	0.476
Recall	0.657	0.641	0.658	0.630	0.617	0.598	0.589	0.598	0.513	0.573
F1 score	0.589	0.576	0.589	0.542	0.556	0.536	0.528	0.536	0.445	0.514
Coverage	4	3	4	2	3	4	4	4	3	4
Leaves	10	6	10	5	6	7	6	7	9	7

Table 4: Evaluation metrics and results.

poral distance (TD) at 2.5 (root) and 0.5 (depth 1). When $TD \in [1, 2]$ (left branch), the model looks at the cognitive status, where “it” is used if an object is in focus (I), “the $\langle N' \rangle$ ” is used otherwise. When $TD = 0$ (i.e., $TD < 0.5$), “the $\langle N' \rangle$ ” is used. When TD is far ($TD \geq 3$), i.e., the right side of the tree, physical distance (PD) is used to differentiate between “this $\langle N' \rangle$ ” and “the $\langle N' \rangle$ ” of $PD \geq 3$. When the objects are closer ($PD \leq 1$, i.e., the objects are in near middle (NM), in hand or in building), cognitive status and temporal distance plays a more important role. Specifically, “that” is used when the cognitive status is in focus and mentioned a few utterances ago ($TD \geq 4$). The number of distractors was not selected as a decision node.

For the eight-class referring form classification, the accuracy score dropped up to approximately 10% to 51.30%–59.83%. M3', without the number of distractors feature, performed as well as full model M1'.

Figure 5 shows the visualization of the M1' model. Because indefinite referring forms were added and they were used to refer to non-present objects, the physical distance feature determines when to use “a $\langle N' \rangle$ ”, as seen in the rightmost traversal. Within the task environment, on the far side, physical distance separates the usage of “this $\langle N' \rangle$ ” and “the N” (the third-right most and the second-right most leaves); this is exactly the same as M1 model, as seen in the rightmost subtree in Figure 4. For the cognitive status, “the $\langle N' \rangle$ ” is

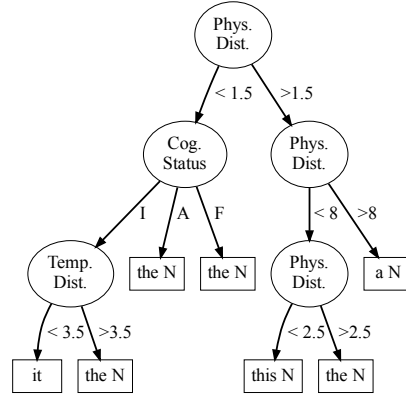


Figure 5: The decision tree visualization for the best-performing M1' model (eight referring form categories). Because indefinite nouns are included, physical distance became the root node.

used when it is lower than In Focus (I) and the object is in front of the instructor ($PD = 1$, i.e., $PD < 1.5$). For In Focus, if the object is less temporally distant ($TD \geq 3$), “it” is used as expected; Otherwise, “the $\langle N' \rangle$ ” is used. When the cognitive status is Activated or Familiar, “the $\langle N' \rangle$ ” is used.

The unpruned trees are available in Appendix 9. As there are many branches and leaves, we do not step through them here. To show the simplicity of the tree with maximum allowed depth 5, Appendix 9 shows the trees with maximum depth 6 and Appendix 9 shows those trees without maximum depth set.

Compared with Pal et al.’s model performance

2021 (72%-86%), the performance of those models trained with the new dataset, especially with the frequently-used indefinite nouns, yielded approximately 20% drop in performance.

7 Discussion

By designing a new task, we were able to collect a situated corpus with a wide variety of referring forms. The corpus also include two frequently used indefinite nouns that were not observed in previous corpora, thanks to careful manipulation of object visibility and the partially observable environment with four quadrants. Using this more ecologically valid task environment, we were able to show that the high performance of Pal et al. (2021)'s work may have been artificially inflated by the nature of their task environment.

Before continuing, we would like to state that this work is not a simple replication of Pal et al.'s paper. Our work contributes a novel situated building-construction task that is much improved over the task used by Pal et al. (2021). Moreover, we expand significantly beyond their work, dealing with more difficult issues such as significantly more objects, their visibility and cognitive status, and ambiguity. In the rest of this section we detail and further interrogate why we believe we observed this performance difference.

First, Pal et al. (2021)'s model was trained on a small dataset of referring forms, in which all have similar cognitive status (activated or in focus) due to the small set of 11 objects (compared to 72 objects in this work). Pal et al. (2021)'s task environment also involved very short dialogues, whereas our tower construction task took an average of half an hour to finish. The small dataset used by Pal et al. (2021) may have resulted in over-fitting.

Second, in Pal et al. (2021)'s task, all objects were either labeled or uniquely distinguishable. In contrast, our tower construction task had only a few shapes of blocks used across 72 blocks, significantly increasing *ambiguity*.

Third, indefinite nouns were not considered by Pal et al. (2021), who only used visible objects. As we see from Figure 3 (the left two bars), indefinite nouns were common in our task. In the previous modeling effort, the cognitive status filters (CSFs) assume all object are at least activated and do not attempt to reason about what is "not known of" to the interlocutor, as the assumption was that both interlocutor and autonomous agents such as robots

know of the same objects in the scene. Future work should weaken this assumption to model Theory of Mind reasoning.

8 Limitations and Future Work

The observed performance gaps motivate possible improvements. During task design, we explicitly intended to collect a multimodal situated dataset, not only language but also gestures, which are particularly informative and suited for situated contexts. We plan to analyze our collected videos and extract gestures, which will likely serve as informative features, as deictic gestures will likely be used on objects' first reference to facilitate use of "this" and "that". In contrast, abstract gestures may be used when objects are in previous quadrants (Stogsdill et al., 2021).

As mentioned in Section 5.2.1, all objects were annotated as at least Familiar to best compare with Pal et al.'s work. Yet, this assumption is clearly violated, especially for objects not yet seen in the task. How to ascribe cognitive status to not-yet-seen objects is a challenging philosophical question, though. We plan to address this in future work.

Finally, to minimize differences between the model trained in this work and that trained by Pal et al. (2021), we excluded a feature that would likely have been informative: referent *visibility*. As discussed, non-visibility was coded as a physical distance of 10; in future work this should be treated as a separate feature.

9 Conclusion

We presented a new interaction-based task design to collect a new situated corpus to advance the computational modelling for referring form selection. Specifically, we adapted the modelling technique used by Pal et al. (2021) and reassess its performance on the new corpus. In future work, we plan to annotate the gestures used in our experiment and improve the computational modelling trained on the new multimodal dataset, moving beyond pure replication.

Supplementary Materials

The data and decision tree code can be found at <https://osf.io/z3ths/>.

Acknowledgements

This work has been supported in part by Office of Naval Research grant N00014-21-1-2418.

References

- Mira Ariel. 1991. The function of accessibility in a theory of grammar. *Journal of pragmatics*, 16(5):443–463.
- Mira Ariel. 2001. Accessibility theory: An overview. *Text representation: Linguistic and psycholinguistic aspects*, 8(8).
- Jennifer E Arnold. 2016. Explicit and emergent mechanisms of information status. *Topics in Cog. Sci.*
- Jennifer E Arnold and Sandra A Zerkle. 2019. Why do people produce pronouns? pragmatic selection vs. rational models. *Language, Cognition and Neuroscience*, 34(9):1152–1175.
- Matthew Aylett and Alice Turk. 2004. The smooth signal redundancy hypothesis: A functional explanation for relationships between redundancy, prosodic prominence, and duration in spontaneous speech. *Language and speech*, 47(1):31–56.
- Maxwell Bennett, Tom Williams, Daria Thames, and Matthias Scheutz. 2017. Differences in interaction patterns and perception for teleoperated and autonomous humanoid robots. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6589–6594. IEEE.
- Susan E Brennan, Marilyn W Friedman, and Carl Pollard. 1987. A centering approach to pronouns. In *ACL*.
- Charles B Callaway and James C Lester. 2002. Pronominalization in generated discourse and dialogue. In *ACL*.
- Frank Eibe, Mark A Hall, and Ian H Witten. 2016. The weka workbench. online appendix for data mining: practical machine learning tools and techniques. In *Morgan Kaufmann*. Elsevier Amsterdam, The Netherlands.
- Victor Ferreira, L Slevc, and Erin Rogers. 2005. How do speakers avoid ambiguous linguistic expressions? *Cognition*, 96(3).
- Michael C Frank and Noah D Goodman. 2012. Predicting pragmatic reasoning in language games. *Science*, 336(6084).
- Gottlob Frege. 1892. Über sinn und bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*, 100:25–50.
- Albert Gatt, Emiel Kraemer, Kees Van Deemter, and Roger Van Gompel. 2014. Models and empirical data for the production of referring expressions. *Lang., Cognition and Neuroscience*.
- Niyu Ge, John Hale, and Eugene Charniak. 1998. A statistical approach to anaphora resolution. In *Workshop on Very Large Corpora*.
- Barbara J Grosz, Aravind K Joshi, and Scott Weinstein. 1995. Centering: A framework for modelling the local coherence of discourse. *Computational Linguistics*.
- André Grüning and Andrej A Kibrik. 2005. Modelling referential choice in discourse: A cognitive calculative approach and a neural network approach. *Anaphora processing: Linguistic, cognitive and computational modelling*, 263:163.
- Jeanette K Gundel, Nancy Hedberg, and Ron Zacharski. 1993. Cognitive status and the form of referring expressions in discourse. *Language*, pages 274–307.
- Nancy Hedberg. 2013. Applying the givenness hierarchy framework: Methodological issues. *International workshop on information structure of Austronesian languages*.
- Rodger Kibble and Richard Power. 2004. Optimizing referential coherence in text generation. *Comp. Ling.*
- Andrej A Kibrik. 2011. *Reference in discourse*. OUP.
- Andrej A Kibrik, Mariya V Khudyakova, Grigory B Dobrov, Anastasia Linnik, and Dmitrij A Zalmanov. 2016. Referential choice: Predictability and its limits. *Frontiers in psychology*, 7:1429.
- Emiel Kraemer and Kees Van Deemter. 2012. Computational generation of referring expressions: A survey. *Computational Linguistics*, 38(1):173–218.
- Olga Krasavina and Christian Chiarcos. 2007. Pocospotdam coreference scheme. In *Linguistic Annotation Workshop*.
- Kathleen F McCoy and Michael Strube. 1999. Generating anaphoric expressions: pronoun or definite description? In *The Relation of Discourse/Dialogue Structure and Reference*.
- Michael Nelson. 2002. Descriptivism defended. *Noûs*, 36(3):408–435.
- Poulomi Pal, Grace Clark, and Tom Williams. 2021. Givenness hierarchy theoretic referential choice in situated contexts. In *Proceedings of the Annual Meeting of the Cognitive Science Society*.
- Poulomi Pal, Lixiao Zhu, Andrea Golden-Lasher, Akshay Swaminathan, and Tom Williams. 2020. Givenness hierarchy theoretic cognitive status filtering. In *Proceedings of the Annual Meeting of the Cognitive Science Society*.
- Massimo Poesio, Rosemary Stevenson, Barbara Di Eugenio, and Janet Hitzeman. 2004. Centering: A parametric theory and its instantiations. *Computational linguistics*, 30(3):309–363.

- JR Quinlan. 1987. Simplifying decision trees. *International journal of man-machine studies*, 27(3):221–234.
- Bertrand Russell. 2001. *The problems of philosophy*. OUP Oxford.
- Adam Stogsdill, Grace Clark, Aly Ranucci, Thao Phung, and Tom Williams. 2021. Is it pointless? modeling and evaluation of category transitions of spatial gestures. In *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, pages 392–396.
- Kees Van Deemter. 2016. *Computational models of referring: a study in cognitive science*. MIT Press.
- Kees Van Deemter, Albert Gatt, Roger PG Van Gompel, and Emiel Krahmer. 2012. Toward a computational psycholinguistics of reference production. *Topics in cognitive science*.
- Ian H Witten and Eibe Frank. 2002. Data mining: practical machine learning tools and techniques with java implementations. *ACM SIGMOD Record*, 31(1):76–77.

Appendix A: Unpruned Decision Trees

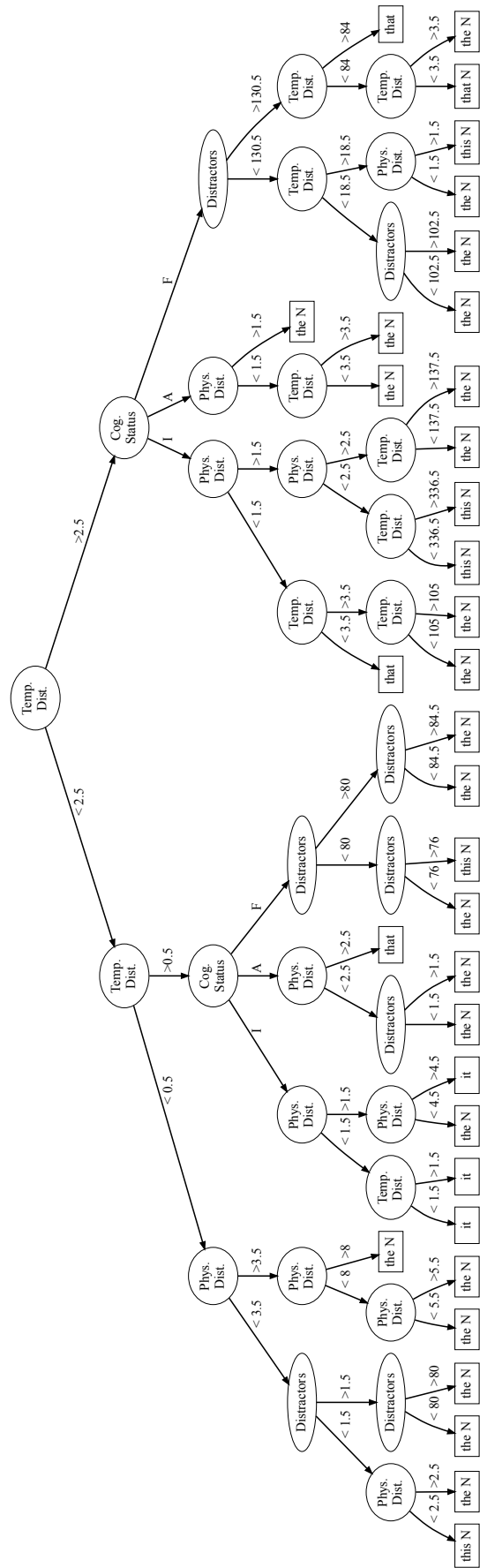


Figure 6: The unpruned decision tree (with six major referring forms).

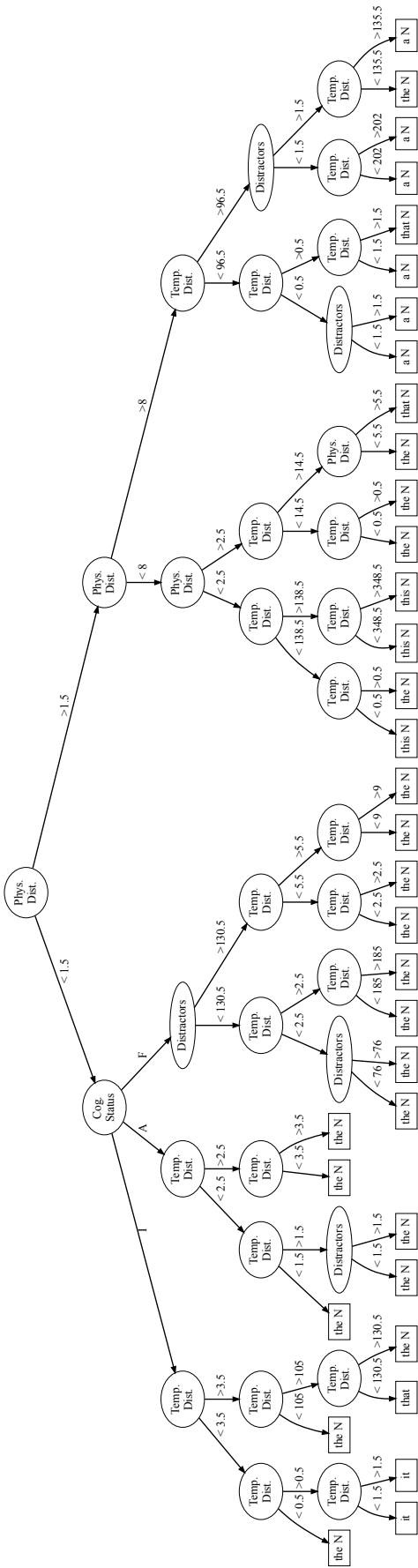


Figure 7: The unpruned decision tree (with six major referring forms and two indefinite referring forms).

Appendix B: Pruned Decision Trees With Maximum Allowed Depth 6

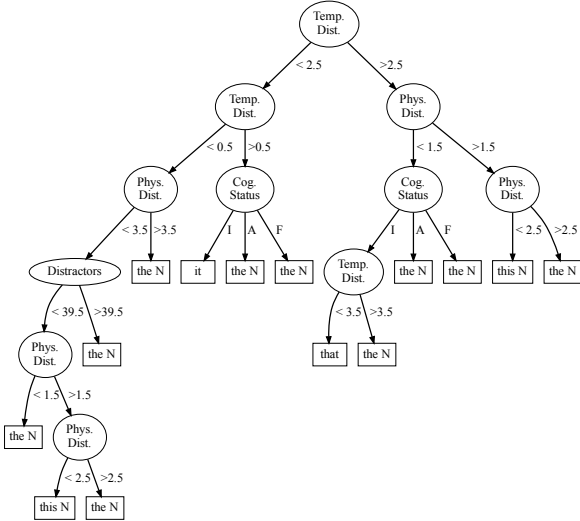


Figure 8: The decision tree visualization for M1 with maximum allowed depth 6 (six main referring form categories informed by Givenness Hierarchy).

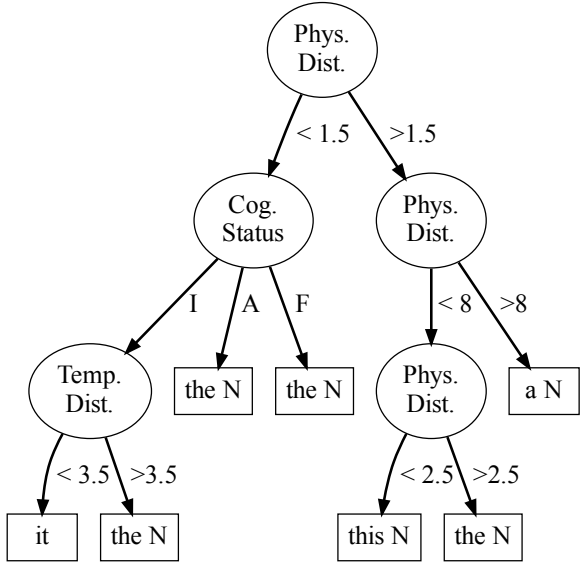


Figure 9: The decision tree visualization for M1 with maximum allowed depth 6 (eight main referring form categories). Note that this is exactly the same as Figure 5.

Appendix C: Pruned Decision Trees With Maximum Allowed Depth Unset

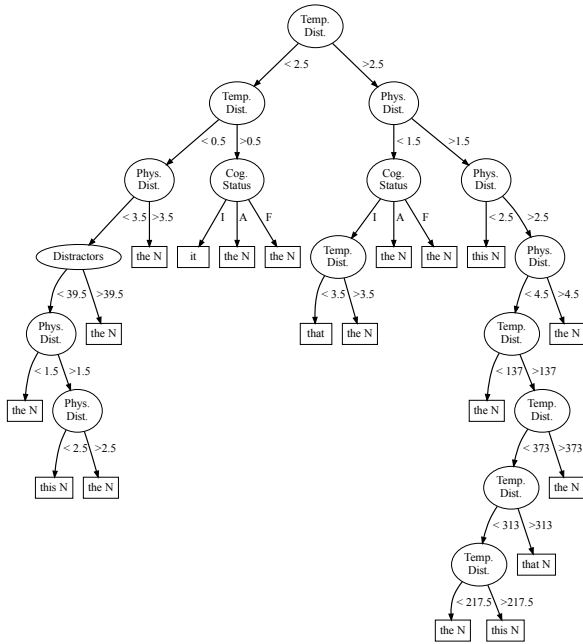


Figure 10: The decision tree visualization for M1 with maximum allowed depth *unset* (six main referring form categories informed by Givenness Hierarchy).

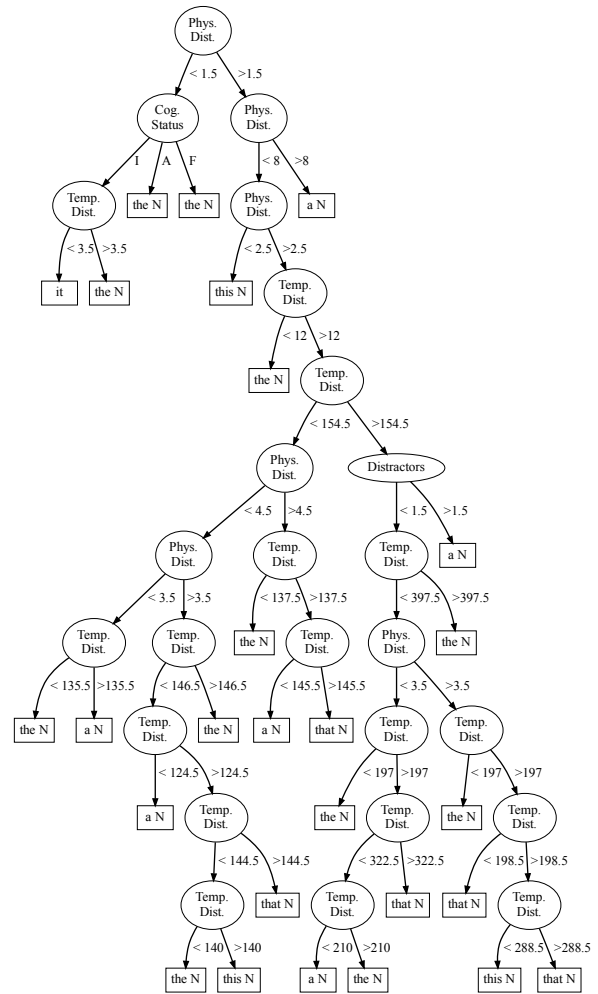


Figure 11: The decision tree visualization for M1 with maximum allowed depth *unset* (eight main referring form categories).

Template-based Approach to Zero-shot Intent Recognition

Dmitry Lamanov¹, Pavel Burnyshev¹, Ekaterina Artemova^{1,2},
Valentin Malykh¹, Andrey Bout¹, Irina Piontkovskaya¹

¹Huawei Noah’s Ark lab, ²HSE University

Correspondence: piontkovskaya.irina@huawei.com

Abstract

The recent advances in transfer learning techniques and pre-training of large contextualized encoders foster innovation in real-life applications, including dialog assistants. Practical needs of intent recognition require effective data usage and the ability to constantly update supported intents, adopting new ones, and abandoning outdated ones. In particular, the generalized zero-shot paradigm, in which the model is trained on the seen intents and tested on both seen and unseen intents, is taking on new importance. In this paper, we explore the generalized zero-shot setup for intent recognition. Following best practices for zero-shot text classification, we treat the task with a sentence pair modeling approach. We outperform previous state-of-the-art f1-measure by up to 16% for unseen intents, using intent labels and user utterances and without accessing external sources (such as knowledge bases). Further enhancement includes lexicalization of intent labels, which improves performance by up to 7%. By using task transferring from other sentence pair tasks, such as Natural Language Inference, we gain additional improvements.

1 Introduction

User intent recognition is one of the key components of dialog assistants. With the advent of deep learning models, deep classifiers have been used throughout to recognize user intents. A common setup for the task (Chen et al., 2019; Wu et al., 2020; Casanueva et al., 2020) involves an omnipresent pre-trained language model (Devlin et al., 2018; Liu et al., 2019b; Sanh et al., 2019), equipped with a classification head, learned to predict intents. However, if the dialog assistant is extended with new skills or applications, new intents may appear. In this case, the intent recognition model needs to be re-trained. In turn, re-training the model requires annotated data, the scope of which is inherently limited. Hence, handling unseen events

defies the common setup and poses new challenges. To this end, **generalized zero-shot (GZS)** learning scenario (Xian et al., 2018), in which the model is presented at the training phase with *seen* intents and at the inference phase with both *seen* and *unseen* intents, becomes more compelling and relevant for real-life setups. The main challenge lies in developing a model capable of processing *seen* and *unseen* intents at comparable performance levels.

Recent frameworks for GZS intent recognition are designed as complex multi-stage pipelines, which involve: detecting unseen intents (Yan et al., 2020), learning intent prototypes (Si et al., 2021), leveraging common sense knowledge graphs (Sid-dique et al., 2021). Such architecture choices may appear untrustworthy: using learnable unseen detectors leads to cascading failures; relying on external knowledge makes the framework hardly adjustable to low-resource domains and languages. Finally, interactions between different framework’s components may be not transparent, so it becomes difficult to trace back the prediction and guarantee the interpretability of results.

At the same time, recent works in the general domain GZL classification are centered on the newly established approach of Yin et al. (2019), who formulate the task as a textual entailment problem. The class’s description is treated as a hypothesis and the text – as a premise. The GZL classification becomes a binary problem: to predict whether the hypothesis entails the premise or not. Entailment-based approaches have been successfully used for information extraction (Haneczok et al., 2021; Lyu et al.; Sainz and Rigau, 2021) and for dataless classification (Ma et al., 2021). However, the entailment-based setup has not been properly explored for GZS intent recognition to the best of our knowledge.

This paper aims to fill in the gap and extensively evaluate entailment-based approaches for GZS intent recognition. Given a meaningful intent label,

such as `reset_settings`, and an input utterance, such as *I want my original settings back*, the classifier is trained to predict if the utterance should be assigned with the presented intent or not. To this end, we make use of pre-trained language models, which encode a two-fold input (intent label and an utterance) simultaneously and fuse it at intermediate layers with the help of the attention mechanism.

We adopt three dialog datasets for GZS intent recognition and show that sentence pair modeling outperforms competing approaches and establishes new state-of-the-art results. Next, we implement multiple techniques, yielding an even higher increase in performance. Noticing that in all datasets considered, most intent labels are either noun or verb phrases, we implement a small set of lexicalizing templates that turn intent labels into plausible sentences. For example, an intent label `reset_settings` is re-written as *The user wants to reset settings*. Such lexicalized intent labels appear less surprising to the language model than intact intent labels. Hence, lexicalization of intent labels helps the language model to learn correlations between inputs efficiently. Other improvements are based on standard engineering techniques, such as hard example mining and task transferring.

Last but not least, we explore two setups in which even less data is provided by restricting access to various parts of annotated data. First, if absolutely no data is available, we explore strategies for transferring from models pre-trained with natural language inference data. Second, in the dataless setup only seen intent labels are granted and there are no annotated utterances, we seek to generate synthetic data from them by using off-the-shelf models for paraphrasing. We show that the sentence pair modeling approach to GZS intent recognition delivers adequate results, even when trained with synthetic utterances, but fails to transfer from other datasets.

The key contributions of the paper are as follows:

1. we discover that sentence pair modeling approach to GZS intent recognition establishes new state-of-the-art results;
2. we show that lexicalization of intent labels yields further significant improvements;
3. we use task transferring, training in dataless regime and conduct error analysis to investigate the strengths and weaknesses of sentence pair modeling approach.

2 Related Work

Our work is related to two lines of research: zero-shot learning with natural language descriptions and intent recognition. We focus on adopting existing ideas for zero-shot text classification to intent recognition.

Zero-shot learning has shown tremendous progress in NLP in recent years. The scope of the tasks, studied in GZS setup, ranges from text classification (Yin et al., 2019) to event extraction (Haneczok et al., 2021; Lyu et al.), named entity recognition (Li et al., 2020) and entity linking (Logeswaran et al., 2019). A number of datasets for benchmarking zero-shot methods has been developed. To name a few, Yin et al. (2019) create a benchmark for general domain text classification. SGD (Rastogi et al., 2020) allows for zero-shot intent recognition.

Recent research has adopted a scope of novel approaches, utilizing natural language descriptions, aimed at zero-shot setup. Text classification can be treated in form of a textual entailment problem (Yin et al., 2019), in which the model learns to match features from class’ description and text, relying on early fusion between inputs inside the attention mechanism. The model can be fine-tuned solely of the task’s data or utilize pre-training with textual entailment and natural language inference (Sainz and Rigau, 2021). However, dataless classification with the help of models, pre-trained for textual entailment only appears problematic due to models’ high variance and instability (Ma et al., 2021). This justifies the rising need for learnable domain transferring (Yin et al., 2020) and self-training (Ye et al., 2020), aimed at leveraging unlabeled data and alleviating domain shift between seen and unseen classes.

Intent recognition Supervised intent recognition requires training a classifier with a softmax layer on top. Off-the-shelf pre-trained language models or sentence encoders are used to embed an input utterance, fed further to the classifier (Casaneva et al., 2020). Augmentation techniques help to increase the amount of training data and increase performance (Xia et al., 2020). Practical needs require the classifier to support emerging intents. Re-training a traditional classifier may turn out resource-greedy and costly. This motivates work in (generalized) zero-shot intent recognition, i.e. handling seen and unseen intents simultaneously. Early approaches to GZS intent recog-

nition adopted **capsule networks** to learn low-dimensional representations of intents. IntentCapsNet (Xia et al., 2018) is built upon three capsule modules, organized hierarchically: the lower module extracts semantic features from input utterances. Two upper modules execute recognition of seen and unseen intents independently from each other. ReCapsNet (Liu et al., 2019a) is built upon a transformation schema, which detects unseen events and makes predictions based on unseen intents’ similarity to the seen ones. SEG (Yan et al., 2020) utilizes **Gaussian mixture models** to learn intent representations by maximising margins between them. One of the concurrent approaches, CTIR (Si et al., 2021) (Class-Transductive Intent Representations) learns **intent representations from intent labels** to model inter-intent connections. CTIR is not a stand-alone solution but rather integrates existing models, such as BERT, CNN, or CapsNet. The framework expands the prediction space at the training stage to be able to include unseen classes, with the unseen label names serving as pseudo-utterances. The current state-of-the-art performance belongs to RIDE (Siddique et al., 2021), an intent detection model that leverages **common knowledge** from ConceptNet. RIDE captures semantic relationships between utterances and intent labels considering concepts in an utterance linked to those in an intent label.

3 Sentence pair modelling for intent recognition

3.1 Problem formulation

Let \mathcal{X} be the set of utterances, $\mathcal{S} = \{y_1, \dots, y_k\}$ be the set of seen intents and $\mathcal{U} = \{y_{k+1}, \dots, y_n\}$ be the set of unseen intents. The training data consists of annotated utterances $\{x_i, y_j\}$. At the test time, the model is presented with a new utterance. In the GZS setup the model chooses an intent from both seen and unseen $y_j \in \mathcal{S} \cup \mathcal{U}$.

3.2 Our approach

A contextualized encoder is trained to make a binary prediction: whether the utterance x_i is assigned with the intent y_j or not. The model encodes the intent description and the utterance, concatenated by the separation token [SEP]. The representation of the [CLS] token is fed into a classification head, which makes the desired prediction $P(1|y_j, x_i)$. This approach follows standard sentence pair (SP) modeling setup.

ID	Template
declarative templates	
d ₁	<i>the user wants to</i> the user wants to book a hotel
d ₂	<i>tell the user how to</i> tell the user how to book a hotel
question templates	
q ₁	<i>does the user want to</i> does the user want to book a hotel
q ₂	<i>how do I</i> how do I book a hotel

Table 1: Lexicalization templates, applied to intent labels. Examples are provided for the intent label “book hotel”.

Given an intent y_j , the model is trained to make a positive prediction for an in-class utterance x_i^+ and a negative prediction for an out-of-class utterance x_i^- , sampled from another intent. At the train time, the model is trained with seen intents only $y_j \in \mathcal{S}$.

On the test time, given an utterance x_i^{test} , we loop over all intents $y_j \in \mathcal{S} \cup \mathcal{U}$ and record the probability of the positive class. Finally, we assign to the utterance x_i^{test} such y^* , that provides with the maximum probability of the positive class:

$$y^* = \arg \max_{y_j \in \mathcal{S} \cup \mathcal{U}} P(1|y_j, x_i^{test})$$

Contextualized encoders. We use RoBERTa_{base} (Liu et al., 2019b) as the main and default contextualized encoder in our experiments, as it shows superior performance to BERT (Devlin et al., 2018) in many downstream applications. RoBERTa’s distilled version, DistilRoBERTa (Sanh et al., 2019) is used to evaluate lighter, less computationally expensive models. Also, we use a pre-trained task-oriented dialogue model, TOD-BERT (Wu et al., 2020) to evaluate whether domain models should be preferred.

We used models, released by HuggingFace library (Wolf et al., 2020): roberta-base, distilroberta-base and TODBERT/TOD-BERT-JNT-V1.

Negative sampling strategies include (i) sampling negative utterances for a fixed intent, denoted as (y_j, x_i^+) , (y_j, x_i^-) ; (ii) sampling negative intents for a fixed utterance, denoted as (y_j^+, x_i) , (y_j^-, x_i) .

Both strategies support sampling with hard examples. In the first case (i), we treat an utterance

x_i^- as a hard negative one for intent y_j , if there exists such in-class utterance x_i^+ , so that the similarity between x_i^+ and x_i^- is higher than a predefined threshold. To this end, to compute semantic similarity, we make use of SentenceBERT (Reimers and Gurevych, 2019) cosine similarity. For a given positive in-class utterance, we selected the top-100 most similar negative out-of-class utterance based on the values of cosine similarity. In the second case (ii), we use the same approach to sample hard negative intents y_j^- , given an utterance x_i , assigned with the positive intent y_j^+ . Again, we compute semantic similarity between intent labels and sample an intent y_j^- with probability based on similarity score with intent y_j^+ . To justify the need to sample hard negative examples, we experiment with random sampling, choosing randomly (iii) negative utterances or (iv) negative intents.

Lexicalization of intent labels utilizes simple grammar templates to convert intent labels into natural-sounding sentences. For this aim, we utilize two types of templates: (i) declarative templates (“*the user wants to*”) and (ii) question templates (“*does the user want to*”). Most intent labels take a form of a verb phrase (VERB + NOUN⁺), such as `book_hotel` or a noun phrase (NOUN⁺), such as `flight_status`. We develop the set of rules that parses an intent label, detects whether it is a verb phrase or a noun phrase¹, and lexicalizes it using one of the templates using the following expression: *template* + VERB + *a/an* + NOUN⁺. If the intent label is recognized as a noun phrase, the VERB slot is filled with an auxiliary verb, “get”. This way, we achieve such sentences: *the user wants to book a hotel* and *does the user want to get a flight status*. The templates implemented are shown in Table 1.

Lexicalization templates were constructed from the most frequent utterance prefixes, computed for all datasets. This way, lexicalized intents sound natural and are close to the real utterances. We use declarative and question templates because the datasets consist of such utterance types. We experimented with a larger number of lexicalization templates, but as there is no significant difference in performance, we limited ourselves to two templates of each kind for the sake of brevity.

Task transferring Task transferring from other tasks to GZS intent recognition allows to estimate

¹We use a basic NLTK POS tagger to process intent labels.

whether (i) pre-trained task-specific models can be used without any additional fine-tuning, reducing the need of annotated data and (ii) pre-training on other tasks and further fine-tuning is beneficial for the final performance.

There are multiple tasks and fine-tuned contextualized encoders, which we may exploit for task transferring experiments. For the sake of time and resources, we did not fine-tune any models on our own, but rather adopted a few suitable models from HuggingFace library, which were fine-tuned on the Multi-Genre Natural Language Inference (MultiNLI) dataset (Williams et al., 2018): BERT-NLI (`textattack/bert-base-uncased-MNLI`), BART-NLI (`bart-large-mnli`), RoBERTa-NLI (`textattack/roberta-base-MNLI`).

Dataless classification We experiment with a dataless classification scenario, in which we train the models on synthetic data. To this end, we used three pre-trained three paraphrasing models to paraphrase lexicalized intent labels. For example, the intent label `get_alarms` is first lexicalized as *tell the user how to get alarms* and then paraphrased as *What’s the best way to get an alarm?*. Next, we merge all sentences, paraphrased with different models, into a single training set. Finally, we train the GZS model with the lexicalized intent labels and their paraphrased versions without using any annotated utterances.

The T5-based (Raffel et al., 2020) and Pegasus-based (Zhang et al., 2020) paraphraser (`ramsrigouthamg/t5_paraphraser`, `Vamsi/T5_Paraphrase_Paws_tuner007/pegasus_paraphrase`) were adopted from the HuggingFace library and were used with default parameters and beam size equal to 25.

4 Datasets

SGD (Schema-Guided Dialog) (Rastogi et al., 2020) contains dialogues from 16 domains and 46 intents and provides the explicit train/dev/test split, aimed at the GZSL setup. Three domains are available only in the test set. This is the only dataset, providing short intent descriptions, which we use instead of intent labels. To pre-process the SGD dataset, we keep utterances where users express an intent, selecting utterances in one of the two cases: (i) first utterances in the dialogue and (ii) an utterance that changes the dialogue state and

Method	SGD				MultiWoZ				CLINC			
	Unseen		Seen		Unseen		Seen		Unseen		Seen	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
SEG	0.372	0.403	0.613	0.636	0.371	0.414	0.652	0.646	-	-	-	-
RIDE+PU	0.590	0.573	0.832	0.830	0.569	0.521	0.884	0.885	0.798	0.573	0.908	0.912
ZSDNN + CTIR	0.603	0.580	0.809	0.878	0.468	0.437	0.827	0.892	0.561	0.493	0.904	0.871
CapsNet + CTIR	0.567	0.507	0.897	0.912	0.481	0.404	0.903	0.906	0.530	0.572	0.866	0.883
SP RoBERTa (ours)	0.698	0.732	0.917	0.925	0.606	0.686	0.903	0.919	0.661	0.742	0.946	0.954
SP RoBERTa + templates (ours)	0.750	0.805	0.931	0.934	0.624	0.722	0.941	0.948	0.692	0.766	0.927	0.931

Table 2: Comparison of different methods. SP stands for Sentence Pair modeling approach. SP RoBERTa (ours) shows consistent improvements of F1 across all datasets for seen and unseen intents. The usage of lexicalized templates improves performance.

expresses a new intent. We use pre-processed utterances from original train/dev/test sets for the GZS setup directly without any additional splitting.

MultiWoZ 2.2 (Multi-domain Wizard of Oz) (Budzianowski et al., 2018) is treated same way as SGD: we keep utterances that express an intent and we get 27.5K utterances, spanning over 11 intents from 7 different domains. We used 8 (out of 11) randomly selected intents as seen for training. 30% utterances from seen intents. All utterances implying unseen intents are used for testing. Test utterances for seen intents are sampled in a stratified way, based on their support in the original dataset.

CLINC (Larson et al., 2019) contains 23,700 utterances, of which 22,500 cover 150 in-scope intents, grouped into ten domains. We follow the standard practice to randomly select 3/4 of the in-scope intents as seen (112 out of 150) and 1/4 as unseen (38 out of 150). The random split was made the same way as for MultiWoZ.

5 Experiments

Baselines We use **SEG**², **RIDE**³, **CTIR**⁴ as baselines, as they show the up-to-date top results on the three chosen datasets. For the RIDE model, we use the base model with a Positive-Unlabeled classifier, as it gives a significant improvement on the SGD and MultiWoZ datasets. We used Zero-Shot DNN and CapsNets along with CTIR, since

²<https://github.com/fanolabs/0shot-classification>, unfortunately were unable to run the code and adopted the published results from the paper

³<https://github.com/RIDE-SIGIR/GZS>

⁴<https://github.com/PhoebusSi/CTIR>

these two encoders perform best on unseen intents (Si et al., 2021).

Evaluation metrics commonly used for the task are accuracy (**Acc**) and **F1**. The F1 values are per class averages weighted with their respective support. Following previous works, we report results on **seen** and **unseen** intents separately. Evaluation for the test set **overall** is presented in Appendix. We report averaged results along with standard deviation for ten runs of each experiment.

Results of experiments are presented in Table 2 (see Appendix for standard deviation estimation). Our approach SP RoBERTa, when used with intent labels and utterances only, shows significant improvement over the state-of-the-art on all three datasets, both on seen and unseen intents, by accuracy and F1 measures. The only exception is unseen intents of CLINC, where our approach underperforms in terms of accuracy of unseen intents recognition comparing to RIDE. At the same time, RIDE shows a lower recall score in this setup. So, our method is more stable and performs well even when the number of classes is high.

Similarly to other methods, our method recognizes seen intents better than unseen ones, reaching around 90% of accuracy and F1 on the former. Next, with the help of lexicalized intent labels our approach yields even more significant improvement for all datasets. The gap between our approach and baselines becomes wider, reaching 14% of accuracy on SGD’s unseen intents and becoming closer to perfect detection on seen intents across all datasets. The difference between our base approach SP RoBERTa and its modification, relying on intent lexicalization, exceeds 7% on unseen in-

Method	SGD		MultiWoZ		CLINC	
	Acc	F1	Acc	F1	Acc	F1
SP RoBERTa	0.687 ± 0.018	0.716 ± 0.016	0.594 ± 0.180	0.705 ± 0.157	0.639 ± 0.038	0.731 ± 0.028
SP BERT	0.668 ± 0.001	0.701 ± 0.001	0.604 ± 0.190	0.704 ± 0.162	0.613 ± 0.023	0.694 ± 0.031
SP TOD-BERT	0.658 ± 0.055	0.724 ± 0.042	0.629 ± 0.235	0.715 ± 0.241	0.625 ± 0.029	0.704 ± 0.034
SP DistilRoBERTa	0.658 ± 0.046	0.710 ± 0.022	0.603 ± 0.208	0.701 ± 0.213	0.583 ± 0.030	0.672 ± 0.029
SP RoBERTa + random IS	0.687 ± 0.018	0.716 ± 0.016	0.594 ± 0.180	0.705 ± 0.157	0.639 ± 0.038	0.731 ± 0.028
SP RoBERTa + random US	0.677 ± 0.017	0.707 ± 0.014	0.531 ± 0.218	0.632 ± 0.217	0.658 ± 0.043	0.735 ± 0.036
SP RoBERTa + hard IS	0.741 ± 0.010	0.786 ± 0.017	0.561 ± 0.177	0.680 ± 0.136	0.590 ± 0.039	0.669 ± 0.036
SP RoBERTa + hard US	0.698 ± 0.012	0.732 ± 0.019	0.606 ± 0.244	0.686 ± 0.234	0.661 ± 0.033	0.742 ± 0.028
Zero-shot RoBERTa-NLI	0.315 ± 0.000	0.382 ± 0.000	0.090 ± 0.000	0.110 ± 0.000	0.065 ± 0.000	0.068 ± 0.000
SP RoBERTa-NLI	0.748 ± 0.026	0.801 ± 0.028	0.669 ± 0.185	0.758 ± 0.151	0.700 ± 0.040	0.771 ± 0.031
SP BERT-NLI	0.693 ± 0.017	0.738 ± 0.015	0.624 ± 0.231	0.715 ± 0.197	0.614 ± 0.035	0.695 ± 0.026
SP BART-NLI	0.789 ± 0.024	0.830 ± 0.030	0.673 ± 0.174	0.753 ± 0.143	0.770 ± 0.039	0.829 ± 0.034

Table 3: Ablation study and task transferring: comparison on unseen intents. **Top**: comparison of different contextualized encoders; **middle**: comparison of negative sampling strategies of intent sampling (IS) and utterance sampling (US); **bottom**: task transferring from the MNLI dataset, using various fine-tuned models.

tents for SGD dataset and reaches 3% on MultiWoZ ones. Notably, SP RoBERTa does not overfit on seen intents and achieves a consistent increase both on unseen and seen intents compared to previous works.

Ablation study We perform ablation studies for two parts of the SP RoBERTa approach and present the results for unseen intents in Table 3. In all ablation experiments we use the SP approach with intent labels to diminish the effect of lexicalization.

First, we evaluate **the choice of the contextualized encoder**, which is at the core of our approach (see the top part of Table 3). We choose between BERT_{base}, RoBERTa_{base}, its distilled version DistilRoBERTa, and TOD-BERT. BERT_{base} provides poorer performance when compared to RoBERTa_{base}, which may be attributed to different pre-training setup. At the same time, TOD-BERT’s scores are compatible with the ones of RoBERTa on two datasets, thus diminishing the importance of domain adaptation. A higher standard deviation, achieved for the MultiWoZ dataset, makes the results less reliable. The performance of DistilRoBERTa is almost on par with its teacher, RoBERTa, indicating that our approach can be used with a less computationally expensive model almost without sacrificing quality.

Second, we experiment with the choice of **negative sampling strategy** (see the middle part of Table 3), in which we can sample either random or hard negative examples for both intents and utterances. The overall trend shows that sampling hard examples improves over random sampling (by up

to 6% of accuracy for the SGD dataset).

Choice of lexicalization templates Table 4 demonstrates the performance of SP RoBERTa with respect to the choice of lexicalization templates. Regardless of which template is used, the results achieved outperform SP RoBERTa with intent labels. The choice of lexicalization template slightly affects the performance. The gap between the best and the worst performing template across all datasets is about 2%. The only exception is q_2 , which drops the performance metrics for two datasets. In total, this indicates that our approach must use just any of the lexicalization templates, but which template exactly is chosen is not as important. What is more, there is no evidence that declarative templates should be preferred to questions or vice versa.

Further adjustments of intent lexicalization templates and their derivation from the datasets seem a part of future research. Other promising directions include using multiple lexicalized intent labels jointly to provide opportunities for off-the-shelf augmentation at the test and train times.

Task transferring results are presented in the bottom part of Table 3. First, we experiment with zero-shot task transferring, using RoBERTa-NLI to make predictions only, without any additional fine-tuning on intent recognition datasets. This experiment leads to almost random results, except for the SGD datasets, where the model reaches about 30% correct prediction.

However, models, pre-trained with MNLI and fine-tuned further for intent recognition, gain sig-

Intent description	SGD		MultiWoZ		CLINC	
	Acc	F1	Acc	F1	Acc	F1
intent labels	0.687 ± 0.018	0.716 ± 0.016	0.594 ± 0.180	0.705 ± 0.157	0.639 ± 0.038	0.731 ± 0.028
d ₁ templates	0.750 ± 0.019	0.805 ± 0.021	0.624 ± 0.231	0.722 ± 0.175	0.692 ± 0.031	0.766 ± 0.028
d ₂ templates	0.752 ± 0.003	0.804 ± 0.006	0.610 ± 0.219	0.713 ± 0.201	0.685 ± 0.035	0.756 ± 0.031
q ₁ templates	0.765 ± 0.019	0.818 ± 0.021	0.621 ± 0.208	0.727 ± 0.174	0.670 ± 0.034	0.747 ± 0.029
q ₂ templates	0.753 ± 0.026	0.807 ± 0.026	0.599 ± 0.212	0.702 ± 0.188	0.554 ± 0.054	0.620 ± 0.055

Table 4: Comparison of different lexicalization templates, improving the performance of SP RoBERTa. Metrics are reported on unseen intents only. Each row corresponds to experiments with a single lexicalization template only, isolated from the others, i.e the row “d₁ templates” uses only the d₁ form.

Train data: intent labels +	SGD				MultiWoZ				CLINC			
	Unseen		Seen		Unseen		Seen		Unseen		Seen	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
original utterances	0.687	0.716	0.916	0.922	0.594	0.705	0.903	0.912	0.639	0.731	0.894	0.903
synthetic utterances	0.666	0.688	0.746	0.778	0.615	0.642	0.621	0.713	0.580	0.613	0.608	0.654

Table 5: Dataless classification. Metrics are reported on seen and unseen intents. Fine-tuning SP-Roberta on synthetic utterances (bottom) shows moderate decline, compared to training on real utterances (top).

nificant improvement up to 7%. The improvement is even more notable in the performance of BART-NLI, which obtains the highest results, probably, because of the model’s size.

Dataless classification results are shown in Table 5. This experiment compares training on two datasets: (i) intent labels and original utterances, (ii) intent labels and synthetic utterances, achieved from paraphrasing lexicalized intent labels. In the latter case, the only available data is the set of seen intent labels, used as input to SP RoBERTa and for further paraphrasing. Surprisingly, the performance declines moderately: the metrics drop by up to 30% for seen intents and up to 10% for unseen intents. This indicates that a) the model learns more from the original data due to its higher diversity and variety; b) paraphrasing models can re-create some of the correlations from which the model learns.

The series of experiments in transfer learning and dataless classification aims at real-life scenarios in which different parts of annotated data are available. First, in zero-shot transfer learning, we do not access training datasets at all (Table 2, Zero-shot RoBERTa NLI). Second, in the dataless setup, we access only **seen** intent labels, which we utilize both as class labels and as a source to create synthetic utterances (Table 5). Thirdly, our main experiments consider both **seen** intents and utterances available (Table 2, SP RoBERTa). In the

second scenario, we were able to get good scores that are more or less close to the best-performing model. We believe efficient use of intent labels overall and to generate synthetic data, in particular, is an important direction for future research.

6 Analysis

Error analysis shows, that SP RoBERTa tends to confuse intents, which (i) are assigned with semantically similar labels or (ii) share a word. For example, an unseen intent `get_train_tickets` gets confused with the seen intent `find_trains`. Similarly, pairs of seen intents `play_media` and `play_song` or `find_home_by_area` and `search_house` are hard to distinguish.

We checked whether errors in intent recognition are caused by utterances’ surface or syntax features. Following observations hold for the SGD dataset. Utterances, which take the form of a question, are more likely to be classified correctly: 93% of questions are assigned with correct intent labels, while there is a drop for declarative utterances, of which 90% are recognized correctly. The model’s performance is not affected by the frequency of the first words in the utterance. From 11360 utterances in the test set, 4962 starts with 3-grams, which occur more than 30 times. Of these utterances, 9% are misclassified, while from the rest of utterances, which start with rarer words, 10% are misclassified.

The top-3 most frequent 3-grams at the beginning of an utterance are *I want to*, *I would like*, *I need to*.

Stress test for NLI models (Naik et al., 2018) is a typology for the standard errors of sentence pair models, from which we picked several typical errors that can be easily checked without additional human annotation. We examine whether one of the following factors leads to an erroneous prediction: (i) word overlap between an intent label and an utterance; (ii) the length of an utterance; (iii) negation or double negation in an utterance; (iv) numbers, if used in an utterance. Additionally, we measured the semantic similarity between intent labels and user utterances through the SentenceBERT cosine function to check whether it impacts performance.

Test	Correct	Incorrect
# overlapping tokens	0.94	0.63
# tokens in utterance	14.96	13.96
# digits in utterance	0.31	0.23
# neg. words in utterance	0.03	0.02
Semantic similarity	0.22	0.21

Table 6: Stress test of SP RoBERTa predictions. An utterance is more likely to be correctly predicted if it shares at least one token with the intent labels.

Table 6 displays the stress test results for one of the runs of SP RoBERTa, trained with q_1 template on the SGD dataset. This model shows reasonable performance, and its stress test results are similar to models trained with other templates. The results are averaged over the test set. An utterance gets more likely to be correctly predicted if it shares at least one token with the intent label. However, the semantic similarity between intent labels and utterances matters less and is relatively low for correct and incorrect predictions. Longer utterances or utterances, which contain digits, tend to get correctly classified more frequently. The latter may be attributed to the fact that numbers are important features to intents, related to doing something on particular dates and with a particular number of people, such as `search_house`, `reserve_restaurant` or `book_appointment`.

7 Conclusion

Over the past years, there has been a trend of utilizing natural language descriptions for various tasks,

ranging from dialog state tracking (Cao and Zhang, 2021), named entity recognition (Li et al., 2020) to the most recent works in text classification employing Pattern-Exploiting Training (PET) (Schick and Schütze, 2020). The help of supervision, expressed in natural language, in most cases not only improves the performance but also enables exploration of real-life setups, such as few-shot or (generalized) zero-shot learning. Such methods’ success is commonly attributed to the efficiency of pre-trained contextualized encoders, which comprise enough prior knowledge to relate the textual task descriptions with the text inputs to the model.

Task-oriented dialogue assistants require the resource-safe ability to support emerging intents without re-training the intent recognition head from scratch. This problem lies well within the generalized zero-shot paradigm. To address it, we present a simple yet efficient approach based on sentence pair modeling, suited for the intent recognition datasets, in which each intent is equipped with a meaningful intent label. We show that we establish new state-of-the-art results using intent labels paired with user utterances as an input to a contextualized encoder and conducting simple binary classification. Besides, to turn intent labels into plausible sentences, better accepted by pre-trained models, we utilized an easy set of lexicalization templates. This heuristic yet alone gains further improvement, increasing the gap to previous best methods. Task transferring from other sentence pair modeling tasks leads to even better performance.

However, our approach has a few limitations: it becomes resource-greedy as it requires to loop over all intents for a given utterance. Next, the intent labels may not be available or may take the form of numerical indices. The first limitation might be overcome by adopting efficient ranking algorithms from the Information Retrieval area. Abstractive summarization, applied to user utterances, might generate meaningful intent labels. These research questions open a few directions for future work.

Acknowledgement

Ekaterina Artemova is supported by the framework of the HSE University Basic Research Program.

References

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ra-

- madan, and Milica Gašić. 2018. [MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.
- Jie Cao and Yi Zhang. 2021. [A comparative study on schema-guided dialogue state tracking](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 782–796, Online. Association for Computational Linguistics.
- Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45.
- Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Jacek Haneczok, Guillaume Jacquet, Jakub Piskorski, and Nicolas Stefanovitch. 2021. Fine-grained event classification in news-like text snippets shared task 2, case 2021. In *Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2021)*, online. Association for Computational Linguistics (ACL).
- Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. [An evaluation dataset for intent classification and out-of-scope prediction](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.
- Jey Han Lau, Carlos Armendariz, Shalom Lappin, Matthew Purver, and Chang Shu. 2020. How furiously can colorless green ideas sleep? sentence acceptability in context. *Transactions of the Association for Computational Linguistics*, 8:296–310.
- Jey Han Lau, Alexander Clark, and Shalom Lappin. 2017. Grammaticality, acceptability, and probability: A probabilistic view of linguistic knowledge. *Cognitive science*, 41(5):1202–1241.
- Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified mrc framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859.
- Han Liu, Xiaotong Zhang, Lu Fan, Xuandi Fu, Qimai Li, Xiao-Ming Wu, and Albert Y.S. Lam. 2019a. [Reconstructing capsule networks for zero-shot intent classification](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4799–4809, Hong Kong, China. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-shot entity linking by reading entity descriptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3449–3460.
- Qing Lyu, Hongming Zhang, Elinor Sulem, and Dan Roth. Zero-shot event extraction via transfer learning: Challenges and insights.
- Tingting Ma, Jin-Ge Yao, Chin-Yew Lin, and Tiejun Zhao. 2021. [Issues with entailment-based zero-shot text classification](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 786–796, Online. Association for Computational Linguistics.
- Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. [Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset](#). In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8689–8696. AAAI Press.

- Nils Reimers and Iryna Gurevych. 2019. Sentencebert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3973–3983.
- Oscar Sainz and German Rigau. 2021. Ask2transformers: Zero-shot domain labelling with pre-trained language models. *South African Centre for Digital Language Resources (SADiLaR) Potchefstroom, South Africa*, page 44.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few-shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*.
- Qingyi Si, Yuanxin Liu, Peng Fu, Zheng Lin, Jiangnan Li, and Weiping Wang. 2021. Learning class-transductive intent representations for zero-shot intent detection. In *IJCAI*.
- AB Siddique, Fuad Jamour, Luxun Xu, and Vagelis Hristidis. 2021. Generalized zero-shot intent detection via commonsense knowledge. *arXiv preprint arXiv:2102.02925*.
- Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.
- Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.
- Chien-Sheng Wu, Steven CH Hoi, Richard Socher, and Caiming Xiong. 2020. Tod-bert: Pre-trained natural language understanding for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 917–929.
- Congying Xia, Caiming Xiong, S Yu Philip, and Richard Socher. 2020. Composed variational natural language generation for few-shot intents. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 3379–3388.
- Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and S Yu Philip. 2018. Zero-shot user intent detection via capsule neural networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3090–3099.
- Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. 2018. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2251–2265.
- Guangfeng Yan, Lu Fan, Qimai Li, Han Liu, Xiaotong Zhang, Xiao-Ming Wu, and Albert YS Lam. 2020. Unknown intent detection using gaussian mixture model with an application to zero-shot intent classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1050–1060.
- Zhiqian Ye, Yuxia Geng, Jiaoyan Chen, Jingmin Chen, Xiaoxiao Xu, SuHang Zheng, Feng Wang, Jun Zhang, and Huajun Chen. 2020. Zero-shot text classification via reinforced self-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3014–3024.
- Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3905–3914.
- Wenpeng Yin, Nazneen Fatema Rajani, Dragomir Radev, Richard Socher, and Caiming Xiong. 2020. Universal natural language processing with limited annotations: Try few-shot textual entailment as a start. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8229–8239.
- Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

A Reproducibility Checklist

A.1 Code

Our code is enclosed in this submission: `gzsl.zip`.

A.2 Computing infrastructure

Each experiment runs on a single NVIDIA V100 16Gb. The longest experiment was running for less than 2.5 hours.

A.3 Datasets

All used datasets are described in the paper. Preprocessing for SGD and MultiWoZ dataset includes (i) selecting utterances from dialogues where users express a new intent, (ii) cleaning uninformative short phrases like acknowledgments and greetings. Preprocessed datasets are also included in `gzsl.zip`. The SGD dataset is released under CC BY-SA 4.0 license. The MultiWoZ dataset is released under Apache License 2.0. To the best of our knowledge the CLINC dataset is released under CC-BY-3.0 license.

A.4 Randomness

All experiments could be reproduced using the fixed set of seeds $\{11..20\}$.

A.5 Evaluation metrics

All used metrics and our motivation to use them are described in the main paper. Metrics and an evaluation script are implemented in our code.

A.6 Models and hyperparameters

Our sentence pair model consists of the contextualized encoder itself, a dropout, and a linear on top of the embedding for [CLS] token. All hyperparameters for the model are fixed in our submission configs. Transformer tokenizers use truncation for utterance and intent description to speed up execution time. Specified values for lexicalized and non-lexicalized setups are reported in `README.md`.

Batch size, learning rate, scheduler, warm-up steps ratio, and other experiment parameters are specified for each dataset and fixed in configs. We used the top 100 out-of-class similar utterances with a positive one as a threshold for hard negative sampling.

A.7 Hyperparameter Search

We performed hyperparameter search using the following grid for each dataset.

- Learning rate: $[2e^{-5}, 5e^{-5}]$
- Batch size: [8, 16]
- Warm up steps ratio: [0.10, 0.15]
- Utterance max length: [20, 30, 40]
- Negative samples k: [5, 7]

For each hyperparameter configuration, we averaged the results over five runs.

A.8 Acceptability evaluation

Lexicalized intent labels help to increase performance since they form more plausible sentences than raw intent labels. This observation can be confirmed by estimating the acceptability of a sentence. We evaluate the acceptability of intent labels and their lexicalized versions with several unsupervised measures, which aim to evaluate to which degree the sentence is likely to be produced (Lau et al., 2017). We exploit the acceptability evaluation tool from (Lau et al., 2020) with default settings. Following acceptability measures have been used: LP stands for unnormalized log probability of the sentence, estimated by a language model. LP_{mean} and LP_{pen} are differently normalized versions of LP with respect to the sentence length. LP_{norm} and $SLOR$ utilize additional normalization with unigram probabilities, computed over a large text corpus. In this experiment, $BERT_{large}$ is used as the default language model; unigram probabilities are pre-computed from bookcorpus-wikipedia. Higher acceptability scores stand for the higher likelihood of the sentence. Thus, more plausible and more natural-sounding sentences gain higher acceptability scores.

We apply one of the lexicalization patterns to all intent labels, score each resulting sentence, and average the achieved scores. Tables 7-9 present with the results of acceptability evaluation for the each dataset. As expected, the intent labels gain lower acceptability scores, while lexicalized patterns receive higher acceptability scores. We may treat the acceptability of the pattern as a proxy to its performance since the $SLOR$ value of the poor performing q_2 pattern is lower than for other patterns.

ID	LP	LP_{mean}	LP_{pen}	LP_{norm}	$SLOR$
labels	-34.58	-18.40	-30.32	-1.84	-8.44
d_1	-43.16	-5.55	-23.50	-0.74	1.97
d_2	-49.92	-5.68	-25.60	-0.77	1.67
q_1	-46.71	-5.31	-23.93	-0.71	2.12
q_2	-43.86	-6.48	-25.48	-0.87	0.98

Table 7: Averaged acceptability scores, computed for the CLINC dataset. Rows stand for intent labels without any changes or lexicalized, using one of the patterns. Higher acceptability scores mean that a sentence is more likely to be grammatical and sound natural. Intent labels less acceptable, while their lexicalized versions form plausible sentences.

ID	LP	LP_{mean}	LP_{pen}	LP_{norm}	$SLOR$
labels	-43.18	-18.45	-36.31	-1.96	-9.01
d_1	-38.93	-5.45	-22.08	-0.72	2.11
d_2	-43.00	-5.29	-22.92	-0.71	2.09
q_1	-41.91	-5.14	-22.32	-0.69	2.31
q_2	-39.36	-6.44	-23.93	-0.86	1.07

Table 8: Acceptability measures, computed for the SGD dataset

ID	LP	LP_{mean}	LP_{pen}	LP_{norm}	$SLOR$
labels	-41.92	-20.96	-37.06	-2.28	-11.77
d_1	-31.45	-4.49	-18.07	-0.62	2.71
d_2	-33.90	-4.24	-18.26	-0.60	2.83
q_1	-33.73	-4.22	-18.17	-0.59	2.93
q_2	-31.87	-5.31	-19.62	-0.75	1.78

Table 9: Acceptability measures, computed for the MultiWOZ dataset

Method	Unseen		Seen		Overall	
	Acc	F1	Acc	F1	Acc	F1
SP RoBERTa + random IS	0.687 ± 0.018	0.716 ± 0.016	0.916 ± 0.005	0.922 ± 0.004	0.884 ± 0.006	0.886 ± 0.005
SP RoBERTa + random US	0.677 ± 0.017	0.707 ± 0.014	0.919 ± 0.005	0.932 ± 0.006	0.885 ± 0.005	0.893 ± 0.005
SP RoBERTa + hard IS	0.741 ± 0.010	0.786 ± 0.017	0.884 ± 0.010	0.891 ± 0.012	0.864 ± 0.009	0.868 ± 0.010
SP RoBERTa + hard US	0.698 ± 0.012	0.732 ± 0.019	0.917 ± 0.003	0.925 ± 0.003	0.887 ± 0.005	0.893 ± 0.008
SP RoBERTa-NLI	0.748 ± 0.026	0.801 ± 0.028	0.923 ± 0.004	0.929 ± 0.003	0.898 ± 0.005	0.905 ± 0.005
SP BERT-NLI	0.693 ± 0.017	0.738 ± 0.015	0.918 ± 0.002	0.924 ± 0.001	0.886 ± 0.003	0.892 ± 0.002
SP BART-NLI	0.789 ± 0.024	0.830 ± 0.030	0.917 ± 0.000	0.924 ± 0.000	0.899 ± 0.003	0.907 ± 0.005
SP RoBERTa + d ₁ patterns	0.750 ± 0.019	0.805 ± 0.021	0.931 ± 0.006	0.934 ± 0.004	0.906 ± 0.004	0.909 ± 0.002
SP RoBERTa + d ₂ patterns	0.752 ± 0.003	0.804 ± 0.006	0.927 ± 0.007	0.932 ± 0.004	0.902 ± 0.005	0.908 ± 0.003
SP RoBERTa + q ₁ patterns	0.765 ± 0.019	0.818 ± 0.021	0.922 ± 0.010	0.927 ± 0.010	0.900 ± 0.007	0.905 ± 0.007
SP RoBERTa + q ₂ patterns	0.753 ± 0.026	0.807 ± 0.026	0.927 ± 0.005	0.931 ± 0.002	0.903 ± 0.004	0.908 ± 0.004

Table 10: Ablation study, task transferring and lexicalization patterns for SGD dataset. **Top**: comparison of negative sampling strategies of intent sampling (IS) and utterance sampling (US); **middle**: task transferring from the MNLi dataset, using various fine-tuned models; **bottom**: Comparison of different lexicalization patterns, improving performance of SP RoBERTa.

Method	Unseen		Seen		Overall	
	Acc	F1	Acc	F1	Acc	F1
SP RoBERTa + random IS	0.594 ± 0.180	0.705 ± 0.157	0.903 ± 0.055	0.912 ± 0.047	0.769 ± 0.082	0.767 ± 0.084
SP RoBERTa + random US	0.531 ± 0.218	0.632 ± 0.217	0.930 ± 0.036	0.938 ± 0.027	0.742 ± 0.096	0.730 ± 0.106
SP RoBERTa + hard IS	0.561 ± 0.177	0.680 ± 0.136	0.937 ± 0.024	0.943 ± 0.016	0.771 ± 0.083	0.761 ± 0.091
SP RoBERTa + hard US	0.606 ± 0.244	0.686 ± 0.234	0.903 ± 0.033	0.919 ± 0.030	0.764 ± 0.099	0.754 ± 0.108
SP RoBERTa-NLI	0.669 ± 0.185	0.758 ± 0.151	0.943 ± 0.014	0.948 ± 0.012	0.808 ± 0.088	0.806 ± 0.089
SP BERT-NLI	0.624 ± 0.231	0.715 ± 0.197	0.941 ± 0.011	0.948 ± 0.010	0.785 ± 0.103	0.782 ± 0.105
SP BART-NLI	0.673 ± 0.174	0.753 ± 0.143	0.946 ± 0.012	0.950 ± 0.010	0.820 ± 0.079	0.814 ± 0.086
SP RoBERTa + d ₁ patterns	0.624 ± 0.231	0.722 ± 0.175	0.941 ± 0.011	0.948 ± 0.010	0.785 ± 0.103	0.782 ± 0.105
SP RoBERTa + d ₂ patterns	0.610 ± 0.219	0.713 ± 0.201	0.944 ± 0.013	0.948 ± 0.011	0.786 ± 0.095	0.781 ± 0.104
SP RoBERTa + q ₁ patterns	0.621 ± 0.208	0.727 ± 0.174	0.946 ± 0.010	0.949 ± 0.010	0.789 ± 0.097	0.786 ± 0.101
SP RoBERTa + q ₂ patterns	0.599 ± 0.212	0.702 ± 0.188	0.943 ± 0.020	0.948 ± 0.015	0.778 ± 0.094	0.775 ± 0.097

Table 11: Ablation study, task transferring and lexicalization patterns for MultiWoZ dataset.

Method	Unseen		Seen		Overall	
	Acc	F1	Acc	F1	Acc	F1
SP RoBERTa + random IS	0.639 ± 0.038	0.731 ± 0.028	0.894 ± 0.009	0.903 ± 0.010	0.768 ± 0.017	0.760 ± 0.017
SP RoBERTa + random US	0.658 ± 0.043	0.735 ± 0.036	0.942 ± 0.007	0.903 ± 0.010	0.791 ± 0.024	0.816 ± 0.019
SP RoBERTa + hard IS	0.590 ± 0.039	0.669 ± 0.036	0.881 ± 0.008	0.901 ± 0.010	0.763 ± 0.020	0.754 ± 0.018
SP RoBERTa + hard US	0.661 ± 0.033	0.742 ± 0.028	0.946 ± 0.007	0.954 ± 0.005	0.794 ± 0.018	0.789 ± 0.020
SP RoBERTa-NLI	0.700 ± 0.040	0.771 ± 0.031	0.950 ± 0.004	0.955 ± 0.003	0.817 ± 0.020	0.836 ± 0.015
SP BERT-NLI	0.614 ± 0.035	0.695 ± 0.026	0.930 ± 0.007	0.938 ± 0.007	0.762 ± 0.020	0.791 ± 0.018
SP BART-NLI	0.770 ± 0.039	0.829 ± 0.034	0.973 ± 0.003	0.976 ± 0.002	0.865 ± 0.022	0.862 ± 0.024
SP RoBERTa + d ₁ patterns	0.692 ± 0.031	0.766 ± 0.028	0.927 ± 0.009	0.931 ± 0.008	0.802 ± 0.018	0.817 ± 0.015
SP RoBERTa + d ₂ patterns	0.685 ± 0.035	0.756 ± 0.031	0.923 ± 0.014	0.928 ± 0.012	0.796 ± 0.024	0.812 ± 0.021
SP RoBERTa + q ₁ patterns	0.670 ± 0.034	0.747 ± 0.029	0.925 ± 0.010	0.930 ± 0.009	0.789 ± 0.019	0.808 ± 0.015
SP RoBERTa + q ₂ patterns	0.554 ± 0.054	0.620 ± 0.055	0.919 ± 0.008	0.921 ± 0.009	0.725 ± 0.029	0.752 ± 0.022

Table 12: Ablation study, task transferring and lexicalization patterns for CLINC dataset.

Train data	SGD						MultiWoZ						CLINC					
	Unseen			Seen			Unseen			Seen			Unseen			Seen		
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1		
intent labels + original utterances	0.687 ± 0.018	0.716 ± 0.016	0.916 ± 0.005	0.922 ± 0.004	0.594 ± 0.180	0.705 ± 0.157	0.903 ± 0.055	0.912 ± 0.047	0.639 ± 0.038	0.731 ± 0.028	0.894 ± 0.009	0.903 ± 0.010						
intent labels + synthetic utterances	0.666 ± 0.019	0.688 ± 0.020	0.746 ± 0.014	0.778 ± 0.014	0.615 ± 0.138	0.642 ± 0.090	0.621 ± 0.101	0.713 ± 0.084	0.580 ± 0.045	0.613 ± 0.040	0.608 ± 0.016	0.654 ± 0.009						

Table 13: Dataless classification. Metrics are reported on seen and unseen intents. Fine-tuning SP-Roberta on synthetic utterances (bottom) shows moderate decline, compared to training on real utterances (top).

Method	SGD						MultiWoZ						CLINC					
	Unseen			Seen			Unseen			Seen			Unseen			Seen		
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1		
SEG	0.372	0.403	0.613	0.636	0.371	0.414	0.652	0.646	-	-	-	-	-	-	-	-		
RIDE+PU	0.590	0.573	0.832	0.830	0.569	0.521	0.884	0.885	0.798	0.573	0.908	0.912						
ZSDNN + CTIR	0.603 ± 0.002	0.580 ± 0.003	0.809 ± 0.006	0.878 ± 0.014	0.468 ± 0.185	0.437 ± 0.176	0.827 ± 0.022	0.892 ± 0.035	0.561 ± 0.059	0.493 ± 0.054	0.904 ± 0.031	0.871 ± 0.026						
CapsNet + CTIR	0.567 ± 0.017	0.507 ± 0.026	0.897 ± 0.010	0.912 ± 0.009	0.481 ± 0.174	0.404 ± 0.243	0.903 ± 0.017	0.906 ± 0.026	0.530 ± 0.049	0.572 ± 0.033	0.866 ± 0.014	0.883 ± 0.020						
SP RoBERTa (ours)	0.698 ± 0.012	0.732 ± 0.019	0.917 ± 0.003	0.925 ± 0.003	0.606 ± 0.244	0.686 ± 0.234	0.903 ± 0.033	0.919 ± 0.030	0.661 ± 0.033	0.742 ± 0.028	0.946 ± 0.007	0.954 ± 0.005						
SP RoBERTa + patterns (ours)	0.750 ± 0.019	0.805 ± 0.021	0.931 ± 0.006	0.934 ± 0.004	0.624 ± 0.231	0.722 ± 0.175	0.941 ± 0.011	0.948 ± 0.010	0.692 ± 0.031	0.766 ± 0.028	0.927 ± 0.009	0.931 ± 0.008						

Table 14: Comparison of different methods. SP stands for Sentence Pair modeling approach. SP RoBERTa (ours) shows consistent improvements of F1 across all datasets for seen and unseen intents. The usage of lexicalized patterns improves performance.

“Slow Service” → “Great Food”: Enhancing Content Preservation in Unsupervised Text Style Transfer

Wanzheng Zhu and Suma Bhat

University of Illinois at Urbana-Champaign, USA
wz6@illinois.edu, spbhat2@illinois.edu

Abstract

Text style transfer aims to change the style (e.g., sentiment, politeness) of a sentence while preserving its content. A common solution is the prototype editing approach, where stylistic tokens are deleted in the “mask” stage and then the masked sentences are infilled with the target style tokens in the “infill” stage. Despite their success, these approaches still suffer from the *content preservation* problem. By closely inspecting the results of existing approaches, we identify two common types of errors: 1) many content-related tokens are masked and 2) irrelevant words associated with the target style are infilled. Our paper aims to enhance content preservation by tackling each of them. In the “mask” stage, we utilize a BERT-based keyword extraction model that incorporates syntactic information to prevent content-related tokens from being masked. In the “infill” stage, we create a pseudo-parallel dataset and train a T5 model to infill the masked sentences without introducing irrelevant content. Empirical results show that our method outperforms the state-of-the-art baselines in terms of content preservation, while maintaining comparable transfer effectiveness and language quality.

1 Introduction

There is growing research interest in text style transfer recently, with the aim of altering the text style (e.g., sentiment, politeness, formality) of a sentence while preserving its content. For example, a sentiment transfer model may transfer a positive-sentiment sentence from “This is the best book I’ve read ever!” to “This is the worst book I’ve read ever!”. As another example, “what happened to my personal station?” may be transferred to “could you please let me know what happened to my personal station?” for a more polite expression. Text style transfer has been shown to be useful in many downstream applications, such as author obfuscation (Shetty et al., 2018), data augmentation (Xie

et al., 2020; Kaushik et al., 2019), text simplification (Xu et al., 2015), and writing assistance (Heidorn, 2000).

Unsupervised style transfer has been extensively explored since parallel data are difficult to obtain. One intuitive and promising solution is the prototype editing approach (Li et al., 2018; Wu et al., 2019; Reid and Zhong, 2021), where the “mask” and “infill” steps are sequentially applied. In the “mask” stage, stylistic tokens are identified and deleted by frequency-ratio based methods (e.g., TF-IDF) and/or attention-based methods, resulting in a content-only masked sentence. In the “infill” stage, the masked sentence is infilled by adding new style markers through template-based methods (Li et al., 2018) or masked language models (Wu et al., 2019; Malmi et al., 2020).

While these models have shown their power to transfer the input text to the target style with high transfer effectiveness, most of them, if not all, suffer from the content preservation issue. As shown in Table 1, despite the style has been transferred successfully, the content is partially changed too (e.g., “service” → “food”).

In this paper, we propose a novel approach to enhance **content preservation** for unsupervised text style transfer. We first summarize two important observations of common errors made by the existing models:

- In the “mask” stage, content-related tokens may be removed (e.g., underlined tokens in cases (a), (c), (d), (e) in Table 1);
- In the “infill” stage, irrelevant words with strong styles may be generated (e.g., underlined tokens in (a), (b), (d), (e) in Table 1).

To preserve content-related tokens in the “mask” stage, we extract the central component of the sentence and prevent them from being masked. Specif-

Transfer Type	Source Sentences	Transferred Sentences
(a) Negative → Positive :	we sit down and we got some really <u>slow and lazy service</u> .	we sit down and we got some really <u>good food and loved it</u> .
(b) Positive → Negative :	the taste is <u>awesome</u> .	the taste is <u>not good and the service is slow</u> .
(c) Factual → Romantic :	a man and a woman show their <u>tattooed</u> hearts on their <u>wrists</u> .	a man and a woman show their <u>loved</u> hearts on their <u>anniversary</u> .
(d) Male → Female :	the locker room is <u>clean</u> .	the locker room is <u>cute</u> .
(e) Toxic → Civil :	as <u>stupid and arrogant</u> as his boss.	as <u>warm hearted</u> as his boss.

Table 1: Error analysis of existing state-of-the-art models. Tokens masked are in red, and new tokens generated are in blue. Tokens underlined are either content-related tokens removed or irrelevant words generated.

ically, we utilize a BERT-based keyword extraction model which incorporates syntactic information (e.g., dependency parsing) to identify content-related tokens. In dependency parsing, the head word of a constituent is the central organizing word of a larger constituent (e.g., the primary noun in a noun phrase, or verb in a verb phrase) (Jurafsky, 2000), and therefore, should be more likely to remain unmasked. Lastly, we make use of an attention network to decide which tokens are stylistic and therefore, should be masked. In style classification tasks, attention scores are often used to interpret to what extent a token has style attribute (Lee et al., 2021; Wu et al., 2019).

In the “infill” stage, existing state-of-the-art approaches typically fine-tune a large pre-trained masked language model (e.g., BERT) on the target style corpus and treat it as a fill-in-the-mask problem (Wu et al., 2019; Malmi et al., 2020; Reid and Zhong, 2021). While such language models can generate fluent sentences of the target style well, they often introduce tokens irrelevant to the source sentence, which results in the change of content. To prevent irrelevant words generation in the “infill” stage, we create a pseudo-parallel dataset and train a large pre-trained language model—T5 (Raffel et al., 2020) to specifically learn to generate from a masked sentence to a target style sentence without introducing unnecessary and irrelevant content.

To summarize, we make the following contributions to enhance content preservation in unsupervised text style transfer:

- In the “mask” stage, we utilize a BERT-based keyword extraction model and leverage dependency parsing information to preserve content-related tokens.
- In the “infill” stage, we propose to create a pseudo-parallel dataset in a self-supervised

manner, and explicitly learn to recover the masked sentences in the target style without adding irrelevant content.

2 Proposed Model

2.1 Problem Formulation

In this paper, we formulate the unsupervised text style transfer as follows: for two non-parallel corpora $\mathbf{X} = \{x_1, x_2, \dots, x_m\}$ and $\mathbf{Y} = \{y_1, y_2, \dots, y_n\}$ with styles S_x and S_y respectively, the goal is to train a style transfer model G that generates a corpus $\hat{\mathbf{X}} = \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_m\}$ conditioned on the corpus \mathbf{X} . The generated corpus $\hat{\mathbf{X}}$ is expected to be in the target style S_y and preserves the content of \mathbf{X} .

2.2 Model Overview

Figure 1 illustrates our proposed model architecture. Following Li et al. (2018); Wu et al. (2019), we assume that style is localized to certain tokens in a sentence and those tokens can be deleted to form a style-free corrupted sentence.¹

At the **training** stage, we first build a style removal model G_d to obtain corrupted sentences \mathbf{Y}_c from \mathbf{Y} , the collection of sentences in the target corpus.² Such corrupted sentences \mathbf{Y}_c are considered style-free under our aforementioned assumption, and ideally there is little loss of content. Second, we train a sentence recovery model G_r to recover the original sentences \mathbf{Y} from the corrupted sentences \mathbf{Y}_c . Such a sentence recovery model G_r is expected to recover the style-free corrupted sentences \mathbf{Y}_c to the original sentences \mathbf{Y} in the target

¹Note that this assumption is not always true. Readers are referred to Jafaritazehjani et al. (2020) for a more detailed discussion.

²“Corrupted sentences” and “masked sentences” are used interchangeably.

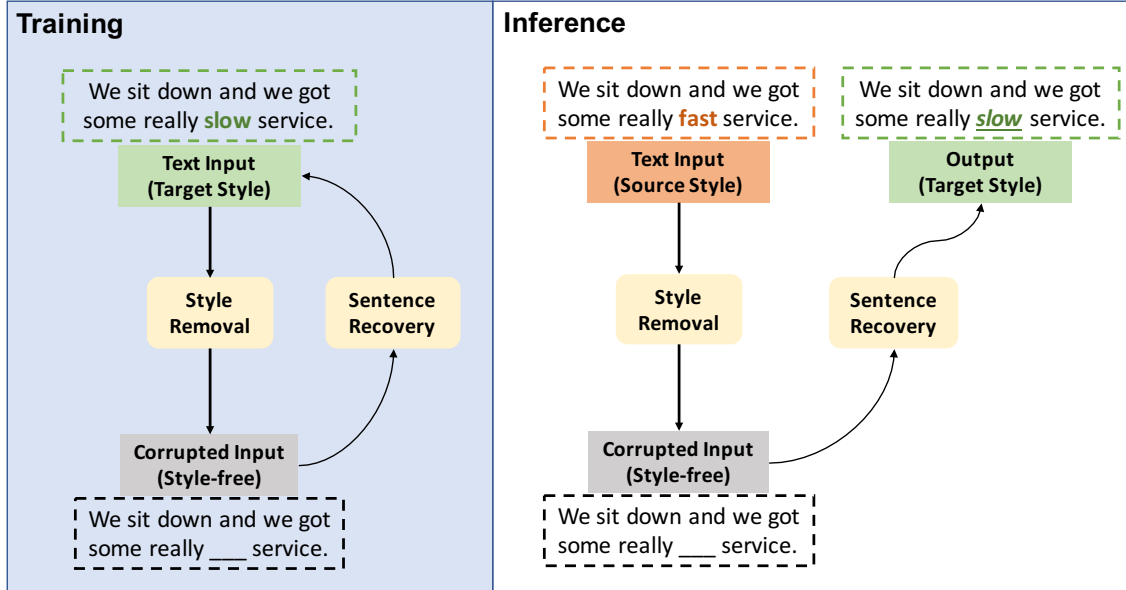


Figure 1: Overview of the model architecture.

style S_y , and very importantly, without introducing irrelevant content.

After training, we have a style removal model G_d and a sentence recovery model G_r . Now at the **inference** stage, we apply the style removal model G_d on the source style sentences \mathbf{X} and obtain style-free corrupted sentences \mathbf{X}_c . Then, we produce the final output $\hat{\mathbf{X}}$ using the sentence recovery model G_r , which is trained to recover corrupted sentences to the target style S_y .

Next, we introduce the details of the style removal model G_d in Section 2.3 and the sentence recovery model G_r in Section 2.4.

2.3 The Style Removal Model

Existing models typically make use of frequency-ratio based methods (e.g., TF-IDF) and/or attention based methods to remove the stylistic tokens (Li et al., 2018; Wu et al., 2019). However, they achieve mediocre performance as many content-related and style-free tokens are masked too. Section 2.3.1 explains how content-related tokens are preserved and Section 2.3.2 shows how the style-related tokens are masked.

2.3.1 Keyword Extraction

To preserve the relevant content, we explicitly utilize a keyword extraction model, which incorporates syntactic information (e.g., dependency parsing) to highlight the content-related tokens and prevent them from removal.

With a source style sentence $x = \{t_1, t_2, \dots, t_k\}$,

where t_i is the i -th token, the model extracts content-related keywords in three steps:

(a) **Embedding**: we use BERT embeddings³ to represent all of the keywords $e_{t_1}, e_{t_2}, \dots, e_{t_k}$ and the entire sentence e_x in a high-dimensional vector space.

(b) **Dependency Parsing**: we construct a dependency tree that captures word-level relations with the Stanford dependency parser (Manning et al., 2014). From the dependency tree, we obtain the depth d_i and the outdegree o_i for each word token t_i . In dependency parsing, the head word of a constituent was the central organizing word of a larger constituent (Jurafsky, 2000). The more central the words are (higher depth or larger outdegree), the more likely it contains meaningful content and therefore, the less likely they should be masked.

(c) **Ranking**: all candidates are ranked to represent the keywords of the sentence:

$$r_{t_i} = \alpha \cdot \cos(e_{t_i}, e_x) + \beta \cdot d_i + \gamma \cdot o_i$$

To alleviate the redundant keywords issue, we follow Bennani-Smires et al. (2018) to use Maximal Marginal Relevance (MMR) (Carbonell and Goldstein, 1998) for diversified candidates by optimizing keyword informativeness with dissimilarity among selected candidates.

Finally, we select candidates over a threshold $thres$ and prevent them from being masked. Em-

³We use “bert-base-uncased” in https://huggingface.co/docs/transformers/model_doc/bert.

pirically, we take $\alpha = 0.8$, $\beta = 0.1$, $\gamma = 0.1$, and $thres = 0.74$, based on the results of the validation data in the Yelp dataset.

2.3.2 Attention

After the keywords have been extracted, we train an attention-based classifier to identify the style-related tokens. We simply encode the sentence and concatenate the forward and the backward hidden states for each word with a bidirectional LSTM. After training, the attention-based classifier is expected to generate attention weights, which capture the style information of each word. For simplicity, we follow Wu et al. (2019) and set the averaged attention value in a sentence as the threshold. Words with attention weights higher than the threshold are viewed as style markers. Note that the content-related keywords identified in Section 2.3.1 are preserved and **not** classified as style markers.

2.4 The Sentence Recovery Model

With style-free corrupted sentences \mathbf{X}_c , we focus on recovering them to the target style S_y . Here, we introduce to solve the problem by creating a pseudo-parallel training dataset and training a model G_r for sentence recovery explicitly. Recall that in Section 2.3, we obtain corrupted sentences \mathbf{Y}_c given the original sentences \mathbf{Y} . Therefore, if we take them in a reverse direction, we then have a parallel training dataset to learn from (i.e., $\mathbf{Y}_c \rightarrow \mathbf{Y}$).

We select T5 (Raffel et al., 2020), a strong pre-trained text-to-text model, as the base architecture, and fine-tune it on the constructed pseudo-parallel dataset. After being trained, the model is expected to take as input a corrupted style-free input \mathbf{Y}_c and generate sentences in the target style without introducing additional irrelevant content. Finally, we apply the trained T5 model on corrupted input \mathbf{X}_c and generate the final output \hat{X} , which is expected to be of the target style S_y .

Intuition: As demonstrated by Wu et al. (2019); Malmi et al. (2020), it is an intuitive idea to treat the “infill” step as a fill-in-the-mask problem, and generate sentences by a fine-tuned masked language model. However, such masked language models (e.g., BERT) are designed to predict tokens for a “mask” and generate sentences with the highest sentence probability. Despite that they are able to generate fluent sentences in the target style, they may introduce tokens that are irrelevant to the source sentence (e.g., case (b) in Table 1) and therefore,

may potentially change the content. Here, what we expect is not a general model for generating a fluent sentence, but rather a specialized model that works only for *sentence recovery without introducing irrelevant content*. Therefore, we construct a pseudo-parallel training dataset and train the model in a supervised manner explicitly for this task. After training on such a dataset, the T5 model is expected to learn specifically to generate sentences in the target style without introducing additional and irrelevant information.

3 Empirical Evaluation

In this section, we empirically evaluate the performance of our proposed approach (denoted as “STEC”⁴) and a set of baseline models. We implemented all models in Python 3.7 and conducted all the experiments on a computer with twenty 2.9 GHz Intel Core i7 CPUs and one GeForce GTX 1080 Ti GPU.

3.1 Datasets

Sentiment Transfer: We use the Yelp dataset and the Amazon dataset (Li et al., 2018), which are business reviews on Yelp and product reviews on Amazon respectively. Each of the dataset consists of two non-parallel corpora with positive and negative sentiments. Each example is labeled as having either positive or negative sentiment.

Captions: The Captions dataset (Gan et al., 2017; Li et al., 2018) has image captions labeled as being factual, romantic or humorous. We focus on the task of converting factual sentences into romantic and humorous ones.

Politeness: The Politeness dataset (Madaan et al., 2020) is produced by filtering through the Enron Email corpus (Klimt and Yang, 2004). We aim to transform the tone of a sentence from impolite to polite.

Detoxification: We employed the largest publicly available toxicity detection dataset to date from “Jigsaw Unintended Bias in Toxicity Classification” Kaggle challenge.⁵ We follow Dale et al. (2021) to obtain non-parallel data, and focus on transferring from toxic to non-toxic.

Dataset statistics are presented in Table 2. For the Yelp, Amazon and Captions datasets, human

⁴short for “Style Transfer with Enhanced Content”

⁵https://www.tensorflow.org/datasets/catalog/civil_comments

Dataset	Style	Train	Valid	Test
Yelp	Positive	270K	2K	500
	Negative	180K	2K	500
Amazon	Positive	277K	985	500
	Negative	278K	1015	500
Captions	Romantic	6K	300	-
	Humorous	6K	300	-
	Factual	-	-	300
Politeness	Polite	219K	28K	-
	Impolite	199K	24K	800
Detoxification	Toxic	150K	5K	10K
	Non-toxic	150K	5K	-

Table 2: Dataset statistics for style transfer tasks.

annotated solutions are also provided for measuring content preservation.

3.2 Baselines

We compare our proposed approach with the following competitive baseline models:

1. CAE: it achieves style transfer from nonparallel text by cross alignment of latent representations (Shen et al., 2017).⁶
2. DRG (Li et al., 2018): this is one of the first successful prototype editing methods. We compare against the full method—delete-retrieve-generate.⁷
3. Mask and Infill (MI) (Wu et al., 2019): the style tokens are first separated from content by masking the positions of sentimental tokens with a fusion model. Then, a masked language model is trained to predict words/phrases conditioned on the context and the target style.
4. Tag and Generate (TAG) (Madaan et al., 2020): it first tags tokens with the original style and/or adds new tags inside a sentence. Then, it conditionally generates the target sentence from the tagged source sentence.⁸
5. NAST (Huang et al., 2021): it first predicts word alignments conditioned on the source sentence, and then generates the transferred sentence with a non-autoregressive decoder. We report results by the model building upon StyTrans (Dai et al., 2019).⁹

⁶<https://github.com/shentianxiao/language-style-transfer>

⁷<https://worksheets.codalab.org/worksheets/0xe3eb416773ed4883bb737662b31b4948/>

⁸<https://github.com/tag-and-generate>

⁹<https://github.com/thu-coai/NAST>

6. RACoLN (Lee et al., 2021): it implicitly removes style at the token level using reverse attention, and fuses content information to style representation using conditional layer normalization.¹⁰

3.3 Evaluation

Following prior work (Madaan et al., 2020; Reid and Zhong, 2021), we evaluate all model outputs along three dimensions: transfer effectiveness, content preservation and language quality.

Transfer effectiveness refers to whether the transferred sentences reveal the target style property. *Content preservation* captures how a sentence maintains its original content throughout the transfer process. *Language quality* measures whether the generated sentences are grammatical, fluent and readable.

3.3.1 Automatic Evaluation

Effectiveness: We follow Reid and Zhong (2021) and train a RoBERTa-base classifier on the training data for the respective dataset. Our evaluation classifier achieves accuracy of 98.0% on Yelp, 84.2% on Amazon, 79.6% on Captions, 88.3% on Politeness, and AUC-ROC of 0.97 on Detoxification. We measure the percentage of the generated sentences classified to be in the target domain by the classifier.

Content Preservation: The standard metric for measuring content preservation is BLEU-self (BL-s) (Papineni et al., 2002) which is compared with respect to the original sentences. However, BLEU scores can measure syntactic content preservation only. Besides, to measure semantic content preservation, we report BERTScore-self (BS-s) (Zhang et al., 2019) against the source sentences. In addition, we report BLEU-reference (BL-r) and BERTScore-reference (BS-r) using the human reference sentences on the Yelp, Amazon and Captions datasets (Li et al., 2018).

Language Quality: We adopt GRUEN (Zhu and Bhat, 2020) to evaluate the language quality.

3.3.2 Human Evaluation

In addition to automatic evaluation, we validate the generated outputs with human evaluation. With each model except CAE, we randomly sample 100 outputs from each dataset.¹¹ Given the target style

¹⁰<https://github.com/MovingKyu/RACoLN>

¹¹We excluded CAE for human evaluation because it performs poorly as determined by the automatic evaluation.

and the original sentence, two annotators (graduate students who are specialized in NLP) are asked to evaluate the model generated sentence with a score range from 1 (Very Bad) to 5 (Very Good) on style transfer accuracy, content preservation, and language quality respectively.

3.4 Results

The automatic evaluation results based on *best-found hyperparameters* are summarized in Table 3. We observe a significant improvement in content preservation scores across various datasets (specifically in the Captions dataset and the Detoxification dataset), highlighting the ability of our model to retain content better than the baseline models. Alongside, we observe comparable performance of our model on transfer effectiveness and language quality across various datasets.

As for the human evaluation, we report the average scores from the 2 annotators in Table 4. We observe that the result mainly conforms with the automatic evaluation. Our model received the highest score on the content evaluation metric, while maintaining comparable score on transfer effectiveness and language quality. Both automatic and human evaluation depict the strength of our proposed model in preserving content.

Among all the baselines, TAG has the best performance consistently in both automatic evaluation and human evaluation, in particular, on the Politeness dataset. This is expected as the “tagger” component is designed to find place for insertion of polite expressions inside a sentence.¹²

For the two state-of-the-art papers that tackle content preservation—RACoLN and NAST, though they perform well on some datasets, the models are not robust across different datasets. Comparably, our approach has consistently good performance and therefore, demonstrates its better generalizability.

3.5 Ablation Study

We compare with the following ablations of STEC and show the results in Figure 2:

1. no-parsing: we exclude the dependency parsing information and use BERT embeddings only to preserve the keywords.

¹²Politeness transfer is slightly different from sentiment transfer, and readers are referred to Madaan et al. (2020) for more detailed discussions.

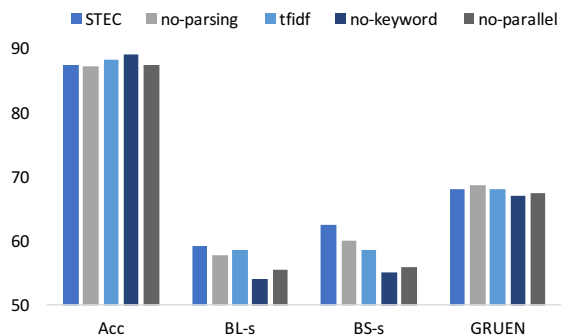


Figure 2: Ablation study. Plots show average results across all five datasets. We scale GRUEN by 100 times for better visualization.

2. tfidf: instead of using the attention network for masking the style-related works, we follow (Li et al., 2018) to use the TF-IDF to mask the style-related words.
3. no-keyword: we exclude the entire keyword extraction model and use the attention network directly to mask the style-related words.
4. no-parallel: instead of constructing a pseudo-parallel dataset and train the T5 model in the “infill” stage, we treat it as a fill-in-the-mask problem and solve it by a fine-tuned masked language model.

We observe that our approach performs better than all ablations in terms of content preservation, and all ablations have comparable performance for transfer effectiveness and language quality. Compared with no-keyword and no-parallel, we conclude that each of the proposed model (i.e., Section 2.3 and Section 2.4) contributes to content preservation well respectively. Besides, by comparing no-keyword and no-parsing, we demonstrate that dependency parsing information can help preserve the content too. In addition, the performance drop by tfidf indicates that an attention network works better in masking stylistic tokens.

3.6 Case Study

Examples of the transferred results by our model are presented in Table 5. We find that our proposed keyword extraction model can preserve the content-related words well. Besides it, we also observe that the T5 model is able to recover the corrupted sentences in the target style without introducing irrelevant content.

	Yelp						Amazon					
	Acc	BL-s	BL-r	BS-s	BS-r	GR	Acc	BL-s	BL-r	BS-s	BS-r	GR
CAE	73.6	20.2	7.7	33.6	22.9	0.69	78.0	2.6	1.7	9.8	6.9	0.51
DRG	88.5	36.7	14.5	48.5	33.3	0.72	51.2	57.1	29.9	66.9	46.2	0.62
MI	90.5	41.7	15.3	49.8	36.0	0.75	74.5	60.0	28.5	61.2	44.7	0.62
TAG	85.8	47.1	19.7	57.9	37.2	0.78	66.4	68.7	34.8	69.5	48.2	0.66
NAST	89.4	59.0	21.0	55.8	45.9	0.72	64.1	55.8	27.9	61.7	39.9	0.59
RACoLN	91.3	58.9	20.0	62.1	42.1	0.75	69.1	31.9	20.1	36.9	31.1	0.63
STEC	88.6	60.2	21.7	62.9	46.6	0.75	66.2	67.1	36.5	68.8	50.9	0.66

(a) Sentiment transfer.

	Captions						Politeness				Detoxification			
	Acc	BL-s	BL-r	BS-s	BS-r	GR	Acc	BL-s	BS-s	GR	Acc	BL-s	BS-s	GR
CAE	89.7	2.1	1.6	11.2	6.7	0.51	99.4	7.0	30.7	0.71	92.3	13.4	22.9	0.52
DRG	95.7	31.8	11.8	40.2	28.4	0.58	90.3	11.8	41.4	0.69	95.6	38.5	42.7	0.58
MI	92.0	42.2	13.3	44.6	31.2	0.64	91.3	55.7	62.9	0.72	95.6	38.9	45.1	0.62
TAG	93.2	51.0	15.6	50.2	36.4	0.65	84.8	70.4	71.6	0.71	92.1	35.1	39.2	0.54
NAST	94.4	44.1	13.3	44.1	32.0	0.64	88.8	65.1	66.7	0.70	93.7	40.1	44.9	0.56
RACoLN	91.2	48.1	13.8	47.7	32.1	0.67	87.5	49.9	54.6	0.71	92.9	36.6	40.3	0.52
STEC	91.5	55.6	17.9	54.8	38.5	0.65	88.9	68.7	71.1	0.71	96.6	42.0	46.1	0.63

(b) Style transfer on other forms.

Table 3: Automatic evaluation results on sentiment transfer. Best results are in bold. Acc: Accuracy; BL-s: BLEU-self; BL-r: BLEU-reference; BS-s: BERTScore-self; BS-r: BERTScore-reference; GR: GRUEN.

	Yelp			Amazon			Captions			Politeness			Detoxification		
	Eff.	CP	LQ	Eff.	CP	LQ	Eff.	CP	LQ	Eff.	CP	LQ	Eff.	CP	LQ
DRG	4.0	3.7	3.5	3.7	3.0	3.3	2.8	2.7	3.0	4.0	4.1	3.7	4.0	3.0	3.3
MI	4.0	3.6	3.7	3.6	2.9	3.4	2.6	3.2	3.1	4.1	4.1	4.0	4.2	2.7	3.0
TAG	3.8	4.0	3.8	3.8	3.4	3.8	3.1	3.5	3.5	4.4	4.5	4.3	3.9	2.6	3.4
NAST	4.2	4.1	3.7	3.4	2.7	3.0	2.2	2.4	2.9	3.9	3.9	3.8	3.9	3.1	3.1
RACoLN	4.3	4.3	3.5	3.1	2.4	3.1	2.4	2.1	2.8	3.7	3.6	3.8	3.6	2.4	2.8
STEC	4.1	4.6	3.7	3.6	3.9	3.6	3.2	3.5	3.2	4.2	4.2	4.0	4.2	3.7	3.4

Table 4: Human evaluation results. Best results are in bold. Eff.: Transfer Effectiveness; CP: Content Preservation; LQ: Language Quality.

4 Related Work

Textual style transfer, the task of changing the style of an input sentence while preserving its content, has recently received increasing attention (Jin et al., 2021). To date, a wide range of solutions have been proposed to solve the task of textual style transfer, such as latent representation disentanglement (Shen et al., 2017; Fu et al., 2018; Riley et al., 2021; Nangi et al., 2021), prototype editing (Li et al., 2018; Wu et al., 2019; Malmi et al., 2020; Madaan et al., 2020; Reid and Zhong, 2021), and others (Gong et al., 2019; Jin et al., 2019; Goyal et al., 2021; Liu et al., 2021).

Many recent works have reported good performance on several aspects of style transfer, including sentiment (Li et al., 2018; Gong et al., 2019), formality (Rao and Tetreault, 2018), simplicity (Van den Bercken et al., 2019; Cao et al., 2020), po-

liteness (Madaan et al., 2020), gender (Prabhumoye et al., 2018), authorship (Jhamtani et al., 2017; Carlson et al., 2018). For instance, Li et al. (2018) propose a simple pipeline approach—delete-retrieve-generate and have shown promising performance on sentiment transfer. Gong et al. (2019) design a reinforcement learning based model for sentiment and formality transfer. It takes style rewards, semantic rewards and fluency rewards from the evaluator and updates the generator for better transfer quality. Madaan et al. (2020) introduce a tag and generate pipeline to identify stylistic words and/or insertion positions. It works particularly well on the Politeness dataset, and shows superior performance on other datasets too.

Content preservation still remains as a major challenge and yet to be solved (Jin et al., 2021; Lee et al., 2021; Huang et al., 2021). To enhance content preservation, researchers have made some

Transfer Type	Source Sentences	Transferred Sentences
(a) Negative → Positive:	we sit down and we got some really slow and lazy service.	we sit down and we got some really great service.
(b) Positive → Negative:	the taste is awesome .	the taste is really bad .
(c) Factual → Humorous:	the group of hikers is resting in front of a mountain.	the group of hikers is being pulled in front of a mountain.
(d) Factual → Romantic:	several young people celebrate by clapping and cheering.	several young people celebrate their lovely friendship by clapping and cheering.
(e) Impolite → Polite:	yes go ahead and remove it	could you please go ahead and remove it
(f) Toxic → Civil:	suggesting that people change their commute times is stupid .	suggesting that people change their commute times is useless .

Table 5: Case study: style transfer results by our proposed model. Tokens masked are in red, and new tokens generated are in blue.

recent progress (Samanta et al., 2021; Garcia et al., 2021; Krishna et al., 2022). For instance, Lee et al. (2021) propose to implicitly remove style at the token level using reverse attention, and fuse content information to style representation using conditional layer normalization. Besides it, Huang et al. (2021) study a non-autoregressive generator, which can serve as an alternative generator for other established models. It explicitly models word alignments to suppress irrelevant words, exploits the word-level transfer between different styles, and is shown to improve content preservation for cycle-loss-based models. In addition, Gong et al. (2020) propose to encode rich syntactic and semantic information with a graph neural network and show its ability on sentiment transfer.

Our work differs from them in the following two aspects: 1) Existing approaches for enhancing content preservation falls in the category of latent representation disentanglement approach, while, to the best of our knowledge, we have proposed the first model to enhance content preserve in the category of prototype editing. 2) Existing approaches rely on the assumption that latent representation can implicitly partially retain both content and style information. However, this assumption lacks justification and remains challengeable (Jin et al., 2021; Jafaritazehjani et al., 2020).

5 Conclusion

In this paper, we identify two common types of errors on content preservation by existing style transfer models. To solve them, we propose to utilize a keyword extraction model to preserve the content-related tokens in the “mask” stage, and to leverage the self-supervision scheme to create a

pseudo-parallel dataset in the “infill” stage. With the two core components, our model is able to enhance content preservation while keeping the outputs with target style. Both automatic and human evaluation shows that our model has strong ability in preserving content and show comparable performance in other evaluation measures too.

Limitation and Future work: 1) we rely on the assumption that style is localized to certain tokens in a sentence and we can delete those tokens to obtain a style-free corrupted sentence. However, this assumption is not always true, especially for more complicated styles (e.g., from modern English to Shakespearean English) (Jafaritazehjani et al., 2020). 2) In more complicated forms of styles, there could be few words associated with the source target, which makes the “mask” model difficult to work well. For instance, in the Politeness dataset, “send me the data” is not a polite expression, but there are no impolite words associated either (Madaan et al., 2020). 3) We focus on the problem of unsupervised style transfer, where access to a large corpus of unpaired sentences with style labels are required. This could be a strong requirement, especially for low-resource settings. Besides, the models built are style-specific and are not generalizable to other styles. It could be an interesting future work to extend our model to the few shot problem setting (Krishna et al., 2022; Garcia et al., 2021).

Acknowledgements

We thank the anonymous reviewers for their helpful comments on earlier drafts that significantly helped improve this manuscript. This study was funded in part by the Jump ARCHES endowment through

the Health Care Engineering Systems Center award number P304.

Ethical Considerations

Risks in deployment: Recent works have highlighted the issues with text style transfer, such as improper usage with malicious intention (Lee et al., 2021) and unintended bias (Krishna et al., 2022). We acknowledge these issues, and given the limited scope of the present study, we call for attention to these aspects by way of well-designed experiments before deployment.

Risks in annotation: The data we use in this paper were posted on publicly accessible websites, and do not contain any personally identifiable information (i.e., no real names, email addresses, IP addresses, etc.). The annotators were warned about the toxic content before they read the data, and were informed that they could quit the task at any time if they were uncomfortable with the content. The annotators in our study were evaluating the quality of the generated sentences only.

References

- Kamil Bennani-Smires, Claudiu Musat, Andreea Hossman, Michael Baeriswyl, and Martin Jaggi. 2018. Simple unsupervised keyphrase extraction using sentence embeddings. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*.
- Yixin Cao, Ruihao Shui, Liangming Pan, Min-Yen Kan, Zhiyuan Liu, and Tat-Seng Chua. 2020. Expertise style transfer: A new task towards better communication between experts and laymen. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in Information Retrieval*.
- Keith Carlson, Allen Riddell, and Daniel Rockmore. 2018. Evaluating prose style transfer with the bible. *Royal Society Open Science*, 5(10):171920.
- Ning Dai, Jianze Liang, Xipeng Qiu, and Xuan-Jing Huang. 2019. Style transformer: Unpaired text style transfer without disentangled latent representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- David Dale, Anton Voronov, Daryna Dementieva, Varvara Logacheva, Olga Kozlova, Nikita Semenov, and Alexander Panchenko. 2021. Text detoxification using large pre-trained neural models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- Chuang Gan, Zhe Gan, Xiaodong He, Jianfeng Gao, and Li Deng. 2017. Stylenet: Generating attractive visual captions with styles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Xavier Garcia, Noah Constant, Mandy Guo, and Orhan Firat. 2021. Towards universality in multilingual text rewriting. *arXiv preprint arXiv:2107.14749*.
- Hongyu Gong, Suma Bhat, Lingfei Wu, JinJun Xiong, and Wen-Mei Hwu. 2019. Reinforcement learning based text style transfer without parallel training corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.
- Hongyu Gong, Linfeng Song, and Suma Bhat. 2020. Rich syntactic and semantic information helps unsupervised text style transfer. In *Proceedings of the 13th International Conference on Natural Language Generation*.
- Navita Goyal, Balaji Vasan Srinivasan, N Anandhavelu, and Abhilasha Sancheti. 2021. Multi-style transfer with discriminative feedback on disjoint corpus. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.
- George Heidorn. 2000. Intelligent writing assistance. *A handbook of natural language processing: Techniques and applications for the processing of language as text*, 8.
- Fei Huang, Zikai Chen, Chen Henry Wu, Qihan Guo, Xiaoyan Zhu, and Minlie Huang. 2021. Nast: A non-autoregressive generator with word alignment for unsupervised text style transfer. In *Findings of the Association for Computational Linguistics (ACL-IJCNLP)*.
- Somayeh Jafaritazehjani, Gwénoné Lecorvé, Damien Lolive, and John Kelleher. 2020. Style versus content: A distinction without a (learnable) difference? In *Proceedings of the 28th International Conference on Computational Linguistics*.
- Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. 2017. Shakespearizing modern language using copy-enriched sequence to sequence models. In *Proceedings of the Workshop on Stylistic Variation*.

- Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. 2021. Deep learning for text style transfer: A survey. *Computational Linguistics*, pages 1–51.
- Zhijing Jin, Di Jin, Jonas Mueller, Nicholas Matthews, and Enrico Santus. 2019. Imat: Unsupervised text attribute transfer via iterative matching and translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.
- Dan Jurafsky. 2000. *Speech & language processing*. Pearson Education India.
- Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations (ICLR)*.
- Bryan Klimt and Yiming Yang. 2004. Introducing the enron corpus. In *CEAS*.
- Kalpesh Krishna, Deepak Nathani, Xavier Garcia, Bidisha Samanta, and Partha Talukdar. 2022. Few-shot controllable style transfer for low-resource multilingual settings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Dongkyu Lee, Zhiliang Tian, Lanqing Xue, and Nevin L Zhang. 2021. Enhancing content preservation in text style transfer using reverse attention and conditional layer normalization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*.
- Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: A simple approach to sentiment and style transfer. In *2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. Association for Computational Linguistics (ACL).
- Yixin Liu, Graham Neubig, and John Wieting. 2021. On learning text style transfer with direct rewards. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.
- Aman Madaan, Amrith Setlur, Tanmay Parekh, Barnabas Póczos, Graham Neubig, Yiming Yang, Ruslan Salakhutdinov, Alan W Black, and Shrimai Prabhumoye. 2020. Politeness transfer: A tag and generate approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Eric Malmi, Aliaksei Severyn, and Sascha Rothe. 2020. Unsupervised text style transfer with padded masked language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd annual meeting of the association for computational linguistics: system demonstrations*.
- Sharmila Reddy Nangi, Niyati Chhaya, Sopan Khosla, Nikhil Kaushik, and Harshit Nyati. 2021. Counterfactuals to control latent disentangled text representations for style transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Association for Computational Linguistics (ACL)*.
- Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.
- Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.
- Machel Reid and Victor Zhong. 2021. Lewis: Levenshtein editing for unsupervised text style transfer. In *Findings of the Association for Computational Linguistics (ACL-IJCNLP)*.
- Parker Riley, Noah Constant, Mandy Guo, Girish Kumar, David C Uthus, and Zarana Parekh. 2021. Textsettr: Few-shot text style extraction and tunable targeted restyling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*.
- Bidisha Samanta, Mohit Agrawal, and Niloy Ganguly. 2021. A hierarchical vae for calibrating attributes

- while generating text using normalizing flow. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*.
- Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. *Advances in neural information processing systems (NIPS)*, 30.
- Rakshith Shetty, Bernt Schiele, and Mario Fritz. 2018. {A4NT}: Author attribute anonymity by adversarial training of neural machine translation. In *27th USENIX Security Symposium*.
- Laurens Van den Bercken, Robert-Jan Sips, and Christoph Lofi. 2019. Evaluating neural text simplification in the medical domain. In *The World Wide Web Conference*.
- Xing Wu, Tao Zhang, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. “Mask and Infill”: Applying masked language model to sentiment transfer. In *International Joint Conference on Artificial Intelligence (IJCAI)*.
- Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems (NIPS)*, 33:6256–6268.
- Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations (ICLR)*.
- Wanzheng Zhu and Suma Bhat. 2020. GRUEN for evaluating linguistic quality of generated text. In *Empirical Methods in Natural Language Processing: Findings (Findings of EMNLP)*.

Arabic Image Captioning using Pre-training of Deep Bidirectional Transformers

Jonathan Emami

Lund University

jontooy@gmail.com

Ashraf Elnagar

University of Sharjah

ashraf@sharjah.ac.ae

Pierre Nugues

Lund University

pierre.nugues@cs.lth.se

Imad Afyouni

University of Sharjah

iafyouni@sharjah.ac.ae

Abstract

Image captioning is the process of automatically generating a textual description of an image. It has a wide range of applications, such as effective image search, auto archiving and even helping visually impaired people to see. English image captioning has seen a lot of development lately, while Arabic image captioning is lagging behind. In this work, we developed and evaluated several Arabic image captioning models with well-established metrics on a public image captioning benchmark. We initialized all models with transformers pre-trained on different Arabic corpora. After initialization, we fine-tuned them with image-caption pairs using a learning method called OSCAR. OSCAR uses object tags detected in images as anchor points to significantly ease the learning of image-text semantic alignments. In relation to the image captioning benchmark, our best performing model scored 0.39, 0.25, 0.15 and 0.092 with BLEU-1,2,3,4 respectively¹, an improvement over previously published scores of 0.33, 0.19, 0.11 and 0.057. Beside additional evaluation metrics, we complemented our scores with human evaluation on a sample of our output. Our experiments showed that training image captioning models with Arabic captions and English object tags is a working approach, but that a pure Arabic dataset, with Arabic object tags, would be preferable.

1 Introduction

The amount of available digital images has increased enormously and captions help us understand and interpret them. While manual captioning is a tedious task, automatic image captioning uses algorithms to extract meaningful information about the content of an image and generate a human-readable sentence from this information.

State-of-the-art automatic image captioning networks are today trained on English corpora. For

¹<https://github.com/jontooy/Arabic-Image-Captioning-using-Transformers>

the other languages, the resulting captions could be translated using a neural machine translation (NMT) model. This procedure, however, introduces an additional source of errors. For Arabic, ElJundi et al. (2020) argued for the necessity of an end-to-end image captioning system that would attenuate errors coming from the unique sentence structure and complex morphology of the Arabic language.

Attai and Elnagar (2020), in a survey on the current state of Arabic image captioning systems, conclude that research conducted for Arabic image captioning is very scarce and that it can mainly be attributed to the lack of publicly available datasets. They also stress that few Arabic image captioning research projects utilized attention mechanisms to focus on the important parts of the image. Such attention mechanisms shall contribute to the caption generation process and give better results.

In their survey, Attai and Elnagar did not mention the transformer architecture as proposed by Vaswani et al. (2017), which is solely based on attention mechanisms. Moreover, transformers in natural language models are gaining more popularity as these models create new state-of-the-art results on different benchmarks, including the OSCAR English image captioning model (Li et al., 2020). This system uses object tags detected in images as anchor points to significantly ease the learning of image-text semantic alignments.

To the best of our knowledge, no transformer-based model for Arabic image captioning had been put to the test. In this paper, we describe an approach to switch the language models of OSCAR with pre-trained Arabic and multilingual ones, then train them on public Arabic benchmark datasets.

The main contributions of this work can be summarized as follows: (i) We evaluate transformer-based Arabic image captioning and compare our results to previous ones. (ii) In relation to the public image captioning benchmark, one of our best per-

forming models scored 0.39, 0.25, 0.15 and 0.092 with BLEU-1,2,3,4 respectively, an improvement over previously published scores of 0.33, 0.19, 0.11 and 0.057. (iii) We show that training image captioning models with Arabic captions and English object tags is a working approach, but that a pure Arabic dataset, with Arabic object tags, is preferable.

2 Related Work

In this section, we summarize recent developments in English image captioning and comment on the current state of Arabic image captioning.

2.1 English Image Captioning

Attention is a technique in neural networks that mimics cognitive attention, and has shown great success in image captioning models ever since [Xu et al. \(2015\)](#) introduced an attention-based model that automatically learns to describe the contents of images. [You et al. \(2016\)](#) developed an algorithm that learns to selectively attend to semantic concept candidates and combine them with hidden states and outputs of recurrent neural networks. [Huang et al. \(2019\)](#) take the attention concept one step further in their work, where they propose an “Attention on Attention” (AoA) module, which extends the conventional attention mechanisms to determine the relevance between attention results and queries.

State-of-the-art image captioning today is based on transformers, an architecture that builds solely on attention mechanisms. [Zhou et al. \(2019\)](#) presented a unified vision-language pre-training (VLP) model which can be fine-tuned for both image captioning and visual question answering (VQA) tasks. [Li et al. \(2020\)](#) presented a new learning method OSCAR (Object-Semantics Aligned Pre-training), and showed that learning of cross-modal representations can be significantly improved by introducing object tags detected in images. These object tags are used as “anchor points” during training to ease the learning of semantic alignments between images and texts. [Zhang et al. \(2021\)](#) studied improved visual representations, dubbed VinVL, and utilized an upgraded approach, dubbed OSCAR+, to pre-train transformer-based VL fusion models. They then fine-tuned the models on various VL benchmarks and created new state-of-the-art results on seven public benchmarks, including image captioning on the COCO Caption benchmark (see

Section 3.1). VinVL has since its release been surpassed by other VLP models, for example LEMON (Large-scale iMAGE captiONer) ([Hu et al., 2021](#)) which studies the scaling behavior of VLP for image captioning.

By the time of this work, VinVL was the state of the art and in this paper, we utilized OSCAR with VinVL on Arabic image captioning.

2.2 Arabic Image Captioning

Arabic image captioning (AIC) introduces additional challenges compared to English captioning. In a survey on the state of AIC, [Attai and Elnagar \(2020\)](#) conclude that research conducted for Arabic image captioning is very scarce and that it can mainly be attributed to the lack of publicly available datasets. The Arabic language is also known for its morphological complexity, and a variety of dialects, which makes it harder to process.

[Jindal](#) leveraged the heavy influence of root words to generate captions of an image directly in Arabic using root word based recurrent neural networks ([Jindal, 2017, 2018](#)). They also reported the first BLEU score for direct Arabic caption generation, from experimental results on datasets from various Middle Eastern newspaper websites and the Flickr8k dataset (see Section 3.2).

[Al-muzaini et al. \(2018\)](#) developed a generative merge model for Arabic image captioning based on a deep RNN-LSTM and a CNN model. They used crowd sourcing to translate samples from two image captioning benchmarks: MS COCO and the Flickr8k dataset. They used a relatively small training set (2400 images) from an unpublished dataset. To reduce the risk of overfitting, [ElJundi et al. \(2020\)](#) developed an annotated dataset for Arabic image captioning (Flickr8k), which, as of today, remains the only public benchmark for AIC. They also developed a base model for AIC that relies on text translation from English image captions and compared it to an end-to-end model that directly transcribes images into Arabic text.

None of the works mentioned above utilized attention mechanisms in their proposed models. [Afyouni et al. \(2021\)](#) developed a hybrid object-based, attention-driven image captioning model. They performed a comprehensive set of experiments using popular metrics and multilingual semantic sentence similarity techniques to assess the lexical and semantic accuracy of generated captions.

Out of all the works from above, only [ElJundi](#)

et al. (2020) have made their dataset publicly available, and is therefore the only work we can directly compare our models with.

When finishing this work, we discovered a Master’s thesis contemporaneous to our work by Sabri (2021). Though not a refereed publication, the author built neural network architectures which include techniques not previously explored in the Arabic image captioning literature, such as transformers. This approach yielded better results over the benchmark published by ElJundi et al. (2020).

3 Datasets

For this work, we mainly used two public datasets for image captioning: Microsoft COCO and Flickr8k. We describe them in detail now.

3.1 Microsoft COCO

Microsoft Common Objects in Context (COCO) (Lin et al., 2014) is a dataset consisting of 123,287 images including object detection, segmentation, and five captions per image (616,435 captions in total). As its name suggests, the COCO dataset contains complex everyday scenes with common objects in their natural context.

For comparison, we adopted the widely used Karpathy split of COCO (Karpathy and Fei-Fei, 2015), i.e. 113,287 train images, 5,000 validation images and 5,000 test images. We used 414,113 pre-translated captions over 82,783 training images with the Advanced Google Translate API², dubbed Arabic-COCO. Figure 1a shows an example of an image from the train split with its five English captions and five Arabic captions. For the Arabic speaking reader, note the error in the second machine translated caption, where the phrase ركوب الأمواج “ride a wave”, should be replaced with its present tense يركب الموج “riding a wave”.

Sabri (2021) showed that, out of a random sampled subset of 150 captions from Arabic-COCO, 46% of the translations were unintelligible. Based on this finding, we considered the captions to be noisy, which is why we did not create a validation and testing set out of Arabic-COCO.

3.2 Flickr8k

The Flickr8k dataset (Hodosh et al., 2013) consists of 8,092 images. Each image in this dataset is associated with five different captions that describe

the entities and events depicted in the image. They were collected via a crowdsourcing marketplace (Amazon Mechanical Turk) with a total of 40,460 captions.

Human translations into Arabic of both the COCO and Flickr8k datasets have been done before. For example, Al-muzaini et al. (2018) built an Arabic dataset based on these two English benchmark datasets. Most of them are not public, therefore we used Arabic Flickr8k by ElJundi et al. (2020). Arabic Flickr8k is split into 6,000 train images, 1,000 validation images, and 1,000 test images, all with three Arabic captions each.

The translation to Arabic was performed by ElJundi et al. in two steps, first by using the Google Translate API and then by validating captions with professional Arabic translators. Finally, they chose the top three translated captions out of five for each image, which makes 24,000 captions in total. Figure 1b shows an example of an image from the train split with its three original English captions and three verified Arabic captions. Note that even though verified, the quality of these Arabic captions is sometimes questionable. For example, the second caption in Figure 1b is رجل أسود, which incorrectly translates to “black man”.

Table 1 shows the complete list of image caption datasets used in this report.

Table 1: Statistics for the Arabic-COCO and Flickr8k translated by ElJundi et al. (2020).

Datasets	Train		Validation		Test	
	#Imgs	#Caps	#Imgs	#Caps	#Imgs	#Caps
Arabic-COCO	82,783	414,113	-	-	-	-
Flickr8k	6,000	18,000	1,000	3,000	1,000	3,000
TOTAL	88,783	432,113	1,000	3,000	1,000	3,000

4 Methodology

As methodology, we used a two-step pipeline, as shown in Figure 2:

1. Extract region features and object tags from an image through a convolutional neural network (CNN) encoder.
2. Generate a sentence from the region features and object tags through a language model, in our case a pre-trained transformer.

As a learning method for our IC model, we used OSCAR (Li et al., 2020) and to evaluate our results, we used well-establish metrics for IC. The following subsections describe these steps in detail.

²<https://github.com/canese-project/Arabic-COCO>



A young boy surfing in low waves.
 A young boy is standing on a surfboard and riding a wave.
 A surfer rides his surf board on some very small waves.
 A young boy is standing on a surfboard in the water.
 A young boy is standing on a surfboard in the ocean.

صبي صغير يتزلج على الأمواج المنخفضة.
 صبي صغير يقف على لوح ركوب الأمواج وركوب الأمواج.
 راكب أمواج يركب لوح الأمواج على بعض الأمواج الصغيرة جدًا.
 صبي صغير يقف على لوح تزلج على الماء في الماء.
 صبي صغير يقف على لوح ركوب الأمواج في المحيط.

(a) COCO



A long-haired man surfing a large wave.
 A man in black on a surfboard riding a wave.
 A man surfing in the ocean.

رجل طويل الشعر يتزلج موجة كبيرة
 رجل أسود على لوح ركوب الأمواج يركب موجة
 رجل يمارس رياضة ركوب الأمواج في المحيط

(b) Flickr8k

Figure 1: Caption annotations in English and Arabic for an image sample from the (a) COCO dataset and the (b) Flickr8k dataset.

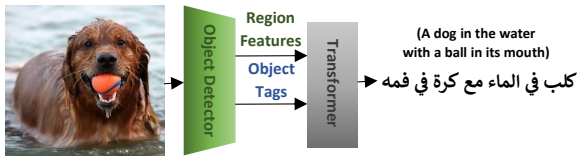


Figure 2: An overview of our methodology.

4.1 Image Feature Extraction and Object Tag Detection

For image feature extraction, Zhang et al. (2021) trained a large-scale object and attribute detection model based on the ResNeXt-152 C4 architecture (Xie et al., 2016), shortened as X152-C4. ResNeXt is named after and adopts the ResNet strategy, a residual learning framework designed to ease the training of networks that are substantially deeper than those used previously (He et al., 2016). For this work, we utilized X152-C4 for feature extraction, pre-trained on 2.49 million unique images, including the COCO dataset. Figure 3 shows an example of object detection with the X152-C4 model. For each detected object, an image region vector is generated, which represents the vector input to the last linear classification layer.

4.2 The Transformer and BERT

The transformer architecture builds solely on attention mechanisms and was first proposed by Vaswani et al. (2017). The transformer has proved



Figure 3: Object detection on an image from the COCO dataset using the X152-C4 architecture. The set of detected object tags are (Arm, Beach, Boy, Cord, Hair, Head, Leaf, Line, Man, Ocean, Person, Sand, Seaweed, Sky, Suit, Surfboard, Tie, Water, Wave, Wetsuit).

superior in sequence-to-sequence modeling, and the key lies in the possibility to capture the relationships between each word in a sequence with every other word.

Proposed by Devlin et al. (2019), BERT showed that pre-trained representations reduced the need for many heavily-engineered task-specific architectures. In other words, by pre-training general language representations, BERT was the first fine-tuning based representation model that achieved

state-of-the-art performance on a large group of sentence-level tasks, outperforming many task-specific architectures.

The release of BERT preceded many other BERT-based language models trained on different corpora in different languages, and will be the main base for our image captioning model. The following paragraphs describe the models used in this work and Table 2 shows the different models configurations for comparison.

mBERT. mBert, short for Multilingual BERT, was pre-trained with the multilingual Wikipedia dataset that consists of the top 104 most common languages (Devlin et al., 2018), including Arabic. In this comparison, we used the `bert-base-multilingual-uncased`³ version of mBERT from HuggingFace.

AraBERT. AraBERT (Antoun et al., 2020) achieved state-of-the-art performance on most tested Arabic NLP tasks. The models were trained on news articles manually scraped from Arabic news websites and several publicly available large Arabic corpora. One of the corpora is named OSCAR (Open Super-large Crawled Aggregated Corpus), not to be confused with the image captioning model OSCAR (Object-Semantics Aligned Pre-training). There are several versions of AraBERT available. We used the `bert-base-arabertv02`⁴ configuration in this work.

ArabicBERT. ArabicBERT (Safaya et al., 2020) was the first pre-trained BERT model for Arabic when it was released. It was originally pre-trained as an approach to solve a sub-task of the Multilingual Offensive Language Identification shared task (OffensEval 2020). We used the `bert-base-arabic`⁵ configuration in this project.

GigaBERT. GigaBERT (Lan et al., 2020) is a set of models pre-trained as a bilingual BERT and designed specifically for Arabic NLP and English-to-Arabic zero-shot transfer learning. Their best model significantly outperforms mBERT and AraBERT on some supervised and zero-shot transfer settings. The training dataset consists of a dump of Arabic Wikipedia, an Arabic version of

OSCAR and the Gigaword corpus, which consists of over 13 million news articles. We used the `GigaBERT-v4-Arabic-and-English`⁶ configuration in this work.

4.3 The OSCAR Learning Method

The vanilla BERT_{BASE} cannot handle image region features as input. As a learning method, we used OSCAR (Li et al., 2020), which achieves state-of-the-art results on six well-established vision-language understanding and generation tasks, including image captioning.

Previous pre-training methods concatenate image region features and text features as input and then use self-attention to learn image-text semantics in a brute force manner. OSCAR uses object tags detected in images as anchor points to ease the alignment of image region and word embeddings. The method is motivated by the observation that the salient objects in an image can be accurately detected by modern object detectors and that these objects are often mentioned in the caption.

The original OSCAR paper adapts pre-trained models to seven downstream VL tasks. For IC fine-tuning, they processed the input samples to triples consisting of image region features, captions, and object tags. They then randomly masked out 15% of the caption tokens and use the corresponding output representations to perform classification and predict the token ids, similar to the masked token loss used by BERT.

We used the caption inference procedure described by Li et al. (2020). They first initialize the caption generation by feeding in a [MASK] token and sampling a token from the vocabulary based on the likelihood of the output. Next, the [MASK] token in the previous input sequence is replaced with the sampled token and a new [MASK] is appended for the next word prediction. The generation process terminates when the model outputs the [STOP] token. We used the same beam search with a beam size of 5.

4.4 Evaluation Metrics

We compared the system performances with evaluation metrics used in machine translation, like BLEU-1,2,3,4 (Papineni et al., 2002), ROUGE-L (Lin, 2004) and METEOR (Banerjee and Lavie, 2005), but also image caption specific metrics³,

³<https://huggingface.co/bert-base-multilingual-uncased>

⁴<https://huggingface.co/aubmindlab/bert-base-arabertv02>

⁵<https://huggingface.co/asafaya/bert-base-arabic>

⁶<https://huggingface.co/lanwuwei/GigaBERT-v4-Arabic-and-English>

³<https://github.com/tylin/coco-caption>

Table 2: Configuration comparisons for mBert, AraBERT, ArabicBERT, and GigaBERT.

Model	Training Data		Vocabulary			Configuration	
	source	#tokens (all/ar)	tokenization	size (all/ar)	cased	size	#parameters
mBERT	Wiki	21.9B/153M	WordPiece	110k/5k	no	base	172M
AraBERT	Wiki, Oscar, News articles	2.5B/2.5B	SentencePiece	64k/58k	no	base	136M
ArabicBERT	Wiki, Oscar	unknown	WordPiece	32k/28k	no	base	111M
GigaBERT	Wiki, Oscar, Gigaword	10.4B/4.3B	WordPiece	50k/26k	no	base	125M

like CIDEr (Vedantam et al., 2014) and SPICE (Anderson et al., 2016). For comparisons of semantic meaning, we utilized the transformer-based Multilingual Universal Sentence Encoder⁴ (MUSE) (Yang et al., 2020) and angular similarity. Specifically, Eq. 1 gives the angular similarity S_θ between two vector embeddings v and u .

$$S_\theta = 1 - \arccos\left(\frac{v \cdot u}{\|v\| \|u\|}\right) / \pi \quad (1)$$

This way of evaluating captions is similar to the technique proposed by Afyouni et al. (2021).

To verify the quality of the candidate captions, we complement our results with human evaluation. For this task, native Arab speaking experts evaluated a sample of the candidate captions generated across the proposed models. We followed the guidelines of the Transparent Human Benchmark (THUMB), a human evaluation protocol proposed by Kasai et al. (2021). The authors base their evaluations on two main scores (*precision* and *recall*) and three types of penalties (*fluency*, *conciseness*, and *inclusive language*).

Precision measures how precise the caption is given the image, while recall measures how much of the salient information (e.g., objects, attributes, and relations) from the image is covered by the caption. Both scores are assessed in the scale of 1–5. The overall score is computed by averaging precision and recall and deducting penalty points, with a maximum deduction of 0.5. Kasai et al. (2021) found most captions from modern neural network models were highly fluent and concise. Since precision and recall covers the context of an image, in our work, the penalty will be purely based on grammar and semantics errors. For example, consider the candidate caption:

فتاة تتارح بمضرب بيسبول على كرة
 “Girl swinging a baseball bat on a ball”

⁴<https://tfhub.dev/google/universal-sentence-encoder-multilingual-large/3>

Although the verb “swinging” is literally translated to تتارح, it does not convey the meaning of the image in Arabic. It should be correctly translated to تضرب “hits” instead, giving the caption 0.5 penalty points.

5 Evaluation

5.1 Preprocessing

Before training the models, we ran all of the images through the X152-C4 object detector for extraction of region features and object tags. Since all of the image features and object tag labels are made available for the Karpathy split of the COCO dataset by Li et al. (2020), only Flickr8k images had to be inferred. We then split the Flickr8k image features and object tags into train, validation, and test images following ElJundi et al. (2020).

To train models on Arabic captions and Arabic object tag *labels*, we simply translated English labels directly with the Google Translate API. A 10% sample of the 1,114 object tags translations detected in the Flickr8k dataset were validated by two native Arab speaking experts on a scale of 1-3 (1: incorrect, 2: partly correct, 3: correct). The annotators gave the sample a mean score of 2.76 and 2.62 with a pairwise Cohen kappa coefficient of 0.43 (moderate agreement).

5.2 Experimental Setup

We initialized the captioning model with various Arabic-specific BERT configurations. In order to select the best models, we carried out two experiments considering the multi/bilingual aspects and the learning curve of the fitting procedure:

1. Evaluation of two multilingual models both trained on
 - (a) Arabic captions and Arabic labels
 - (b) Arabic captions and English labels

We carried out this experiment mainly for comparing the object labels ability to affect the final image-text alignment.

- Evaluation of the learning curve for three different models, respectively trained on 50%, 75% and 100% of a dataset. From the results, we can tell if the validation loss decreases with the amount of data or if some adjustment have to be made to the models, for example with a hyper parameter grid search. Out of the trained models, we chose the two most accurate ones as candidates for large scale training.

After we picked two candidate models, we made a third and final experiment:

- Do large scale training on the candidate models on datasets of different size. Evaluate the models both with automatic and human metrics and compare the results with previous models.

We carried out the first two experiments on Google Colab GPU:s (1 P100 GPU with 16 GB memory). We carried out the final large scale experiments on a workstation (1 GV100 GPU with 32 GB memory) and a high performance computer (HPC) system (8 K80 GPU:s with 12 GB memory each).

For all the experiments above, we saved training and validation loss values at every epoch, while model checkpoints were saved every 5 epochs. All the experiments used the AdamW optimizer and a linearly decaying learning rate according to the recipe described in OSCAR (Li et al., 2020). Exact model hyper parameters for each experiment are shown in Appendix A.

5.3 Experimental Results

English vs Arabic labels. Table 3 shows the final evaluation scores for all models. Our first experiments show that both approaches, training on English and Arabic object labels, work in principle. Already at this stage, GigaBERT trained on English labels outperformed previous reported BLEU-1,2,3,4 scores with 0.0123, 0.0144, 0.0190, 0.0167 respectively. However, note that these scores were obtained from the val-split, and not the final test-split. We think that the reason to why GigaBERT with English labels outperforms Arabic labels is that the quality of the original English labels, in combination with GigaBERT’s English pre-training, is much better than its machine translated counterpart. mBert is only trained on Wikipedia (Devlin et al., 2018), while GigaBERT is trained

on the Gigaword corpus in addition to Wikipedia and web crawl data. This is how we explain GigaBERT’s better performance. Moreover, the vocabulary of GigaBERT (21k English tokens vs 26k Arabic tokens) is richer and more balanced than the vocabulary of mBERT (53k English tokens vs 5k Arabic tokens), see Table 2.

Table 3: Evaluation scores (evaluation on epoch 30) for the trained models. The best scoring models are marked in bold for each evaluation metric.

Model	Labels	BLEU-4	ROUGE-L	METEOR	CIDEr	SPICE
GigaBERT	English	0.074	0.29	0.3	0.33	0.037
	Arabic	0.062	0.29	0.31	0.31	0.037
mBert	English	0.058	0.28	0.30	0.29	0.031
	Arabic	0.067	0.29	0.30	0.31	0.033

Learning Curve. We evaluated all the models from the learning curve experiment with MUSE to investigate the correlation between semantic scores and an increased amount of data. The evaluation over training time is shown in Figure 4 for AraBERT, ArabicBERT, and GigaBERT. In general, more data increased evaluation scores. One notable thing is that the final score of GigaBERT trained on 75% of data outperformed 100%, but Figure 4b shows that the 100% curve is generally higher than the 75% curve. This finding suggests that the average MUSE score has a high variance. Note that GigaBERT trained on 100% of Flickr8k is identical to the model trained on Arabic labels in the previous experiment.

In the case of AraBERT, the 75% MUSE curve is way lower than the 100% and 50% curves, but the 100% loss curve is still higher than the 50% one. The unstable training results of AraBERT suggest that the selected learning rate is too large. We performed learning rate grid search on AraBERT and GigaBERT on the interval $\eta \in [1e^{-5}, 7e^{-5}]$ to minimize validation loss, and found an optimum at $\eta = 3e^{-5}$.

Large Scale Training. Table 4 presents the final test scores (BLEU-1,2,3,4, ROUGE-L, METEOR, CIDEr and MUSE) of a selection of our models, and models previously proposed by Jindal (2018), Al-muzaini et al. (2018), Afyouni et al. (2021) and ElJundi et al. (2020). Out of the previous works, only the model by ElJundi et al. (2020) is tested on the same Flickr8k test set as ours. We were unable to obtain the splits from the other studies, and have no data regarding on how their splits may differ from ours. The difference between their model scores and our are quite large in some cases. One

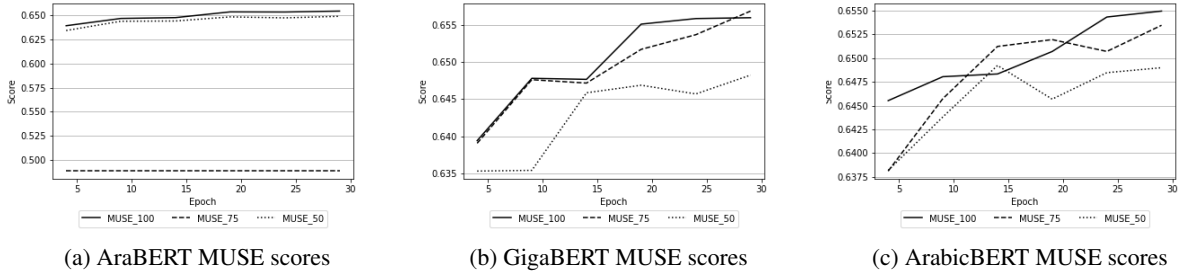


Figure 4: MUSE evaluation scores over all epochs for (a) AraBERT, (b) GigaBERT and (c) ArabicBERT.

Table 4: Our model scores compared to previous models. The highest scores on our test-split are marked in bold. Of all the previous ones, only the model by [ElJundi et al. \(2020\)](#) uses the same test-split as us. Other test-splits are unknown.

Model	Test set	B1	B2	B3	B4	ROUGE-L	METEOR	CIDEr	MUSE
Jindal (2018)	Flickr8k	0.658	0.559	0.404	0.223	-	0.201	-	-
Al-muzaini et al. (2018)	COCO & Flickr8k	0.462	0.260	0.190	0.080	-	-	-	-
Afyouni et al. (2021)	COCO	0.649	0.413	0.241	0.136	0.470	0.408	-	0.78
ElJundi et al. (2020)	Flickr8k	0.332	0.193	0.105	0.057	-	-	-	-
AraBERT32-Flickr8k	Flickr8k	0.391	0.246	0.150	0.092	0.331	0.314	0.415	0.671
AraBERT32-COCO		0.365	0.221	0.129	0.0715	0.310	0.317	0.36	0.669
AraBERT256-Flickr8k		0.387	0.244	0.151	0.093	0.334	0.312	0.428	0.668
GigaBERT32-Flickr8k		0.386	0.241	0.144	0.0827	0.331	0.315	0.403	0.669
GigaBERT32-COCO		0.36	0.215	0.124	0.0708	0.308	0.311	0.344	0.668
			Δ 0.059 \uparrow	0.053 \uparrow	0.046 \uparrow	0.036 \uparrow			

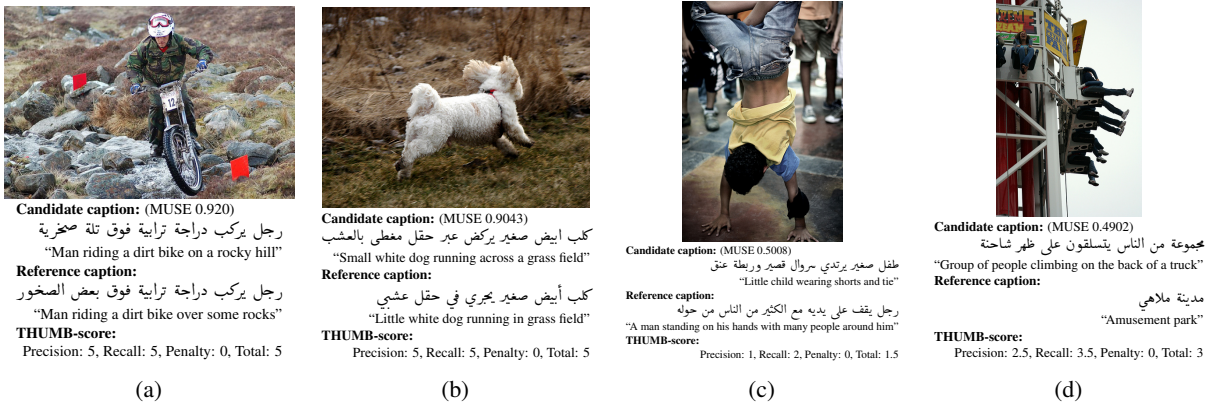


Figure 5: Human evaluation of four candidate captions produced by AraBERT32-COCO: two accurate candidate captions (a) and (b), and two inaccurate candidate captions (c) and (d). Each candidate caption is accompanied by the reference caption from the Flickr8k test-split with the most MUSE similarity, and a THUMB score.

possible explanation could be that our BERT-based approach differs from previous LSTM-based approaches, which can achieve significantly higher results than a BERT-based model for a small dataset on NLP tasks ([Ezen-Can, 2020](#)).

All of our models are named after the scheme *modelBatchSize-dataset*, where *model* is our initialization model, *BatchSize* is the training batch size and *dataset* is the dataset trained on. For example, one of our best performing models was initialized on AraBERT and trained with a batch size of 32 on Flickr8k. Therefore, we named the model

AraBERT32-Flickr8k. AraBERT32-Flickr8k outperforms the model by [ElJundi et al. \(2020\)](#) on all BLEU scores, and most remarkably on BLEU-4, where we see a 61.4% increase. We chose to drop the SPICE scores from Table 4 because of the evaluation scripts incompatibility with the Arabic language.

We complemented Table 4 with human evaluations on a sample of the dataset according to the guidelines of THUMB ([Kasai et al., 2021](#)). Figure 5 shows four generated captions from AraBERT32-COCO with images and human evaluations. All of

the evaluations were made by two Arabic language experts.

In general, the human evaluations show accurate results. In Figure 5a, the candidate caption:

رجل يرك دراجة ترابية فوق تلة صخرية
“Man riding a dirt bike on a rocky hill”

is nearly perfect. It is almost identical to the reference caption:

رجل يركب دراجة ترابية فوق بعض الصخور
“Man riding a dirt bike over some rocks”,

and only differs in the last phrase.

Not all results were accurate. Looking at Figure 5c, the first row shows the candidate caption

مجموعة من الناس يتسلقون على ظهر شاحنة
“Group of people climbing on the back of a truck”,

while the closest reference caption مدينة ملاهي translates to “Amusement park”. Though the candidate sentence is fluent and grammatically correct, it appears to be random in the context of the image. This shows how the models in these examples fail to identify objects in the image and correctly describe a scene.

A potential source of error for the incorrect image-text alignment could be noise in the machine translated data input. For example, the publicly available Arabic-COCO used is purely machine translated and has to be verified by humans before employed in testing. The justification to why we still use machine-translated data is that we rely on the BERT-based language models to handle the grammar and syntax, while we count on the machine-translation model to correctly translate salient objects. The failure to do so leads to errors in learning image-text semantic alignments. For example, in our dataset, mistranslated object labels can be found. Some nouns are mistranslated into their homophone counterparts: “light” (*noun*) to خفيفة (*adjective*, bright; well-lighted), “block” (*noun*) to منع (*adjective*, to obstruct, or prevent someone or something) and so on. Li et al. (2020) showed that OSCAR learning curves for fine-tuning with object tags converge significantly faster than the methods without tags. In other words, high quality labels are crucial in image-text alignment for VL-pretrained models.

For the complete table with scores for all trained models, see Appendix B.

6 Conclusion

This work focused on Arabic image captioning using pre-trained bidirectional transformers. We can draw many conclusions from it.

The specific challenge in Arabic image captioning is, not regarding the lack of well-annotated datasets, the morphological complexity of the Arabic language which makes it harder to process. In our work, we showed that it is possible to achieve state-of-the-art results with a minimal pre-processing scheme and by adapting English captioning models to other languages through public dataset benchmarks.

Furthermore, we achieved results better than the previous work on the Flickr8k dataset by ElJundi et al. (2020). Our experiments also show that both approaches, training on English and Arabic object labels, work in principle. In addition, we proposed working configurations and heuristics for hyper parameters in future experimentation on our proposed models. Therefore, our models provide a new baseline for the AIC community.

Further work in the field should be to verify all machine translated Arabic labels by humans before further training on the datasets. This task should not be too expensive since there are only 1,114 object tags translations detected in the Flickr8k dataset, and 253 additional object tags in Arabic-COCO. This could greatly improve training. Secondly, the lack of qualitative Arabic data should be solved by translation and verification of all COCO captions, and then making the resulting dataset publicly available. As a suggestion, one could follow a crowd sourcing procedure as described by Almuzaini et al. (2018), which includes some of the instructions that were used in the creation of COCO captions, and additional instructions specific to the Arabic language. This would create a new benchmark Arabic captioning dataset that we could train and test our models on.

Finally, we hope that our work will be useful for future Arabic image captioning models, and that it will spur more contributions to the field in the closest future.

Acknowledgements

This work was partially supported by Vetenskaprådet, the Swedish Research Council, registration number 2021-04533.

References

- Imad Afyouni, Imtinan Azhara, and Ashraf Elnagar. 2021. AraCap: A hybrid deep learning architecture for Arabic Image Captioning. In *ACLing 2021: 5th International Conference on AI in Computational Linguistics*.
- Huda A. Al-muzaini, Tasniem N. Al-yahya, and Hafida Benhidour. 2018. Automatic Arabic image captioning using RNN-LSTM-based language model and CNN. *International Journal of Advanced Computer Science and Applications*, 9(6).
- Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. SPICE: Semantic Propositional Image Caption Evaluation. In *Computer Vision – ECCV 2016*, pages 382–398, Cham. Springer International Publishing.
- Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.
- Anfal Attai and Ashraf Elnagar. 2020. A survey on Arabic Image Captioning Systems Using Deep Learning Models. In *14th International Conference on Innovations in Information Technology (IIT)*, pages 114–119.
- Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Multilingual bert readme. <https://github.com/google-research/bert/blob/master/multilingual.md>. [Online; accessed 6 Feb. 2022].
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Obeida ElJundi, Mohamad Dhaybi, Kotaiba Mokadam, Hazem Hajj, and Daniel Asmar. 2020. Resources and End-to-End Neural Network Models for Arabic Image Captioning. In *15th International Conference on Computer Vision Theory and Applications*.
- Aysu Ezen-Can. 2020. A Comparison of LSTM and BERT for Small Corpus. *ArXiv*, abs/2009.05451.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.
- Micah Hodosh, Peter Young, and Julia Hockenmaier. 2013. Framing Image Description as a Ranking Task: Data, Models and Evaluation Metrics. In *24th International Joint Conference on Artificial Intelligence*.
- Xiaowei Hu, Zhe Gan, Jianfeng Wang, Zhengyuan Yang, Zicheng Liu, Yumao Lu, and Lijuan Wang. 2021. Scaling Up Vision-Language Pre-training for Image Captioning. *CoRR*, abs/2111.12233.
- Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. 2019. Attention on Attention for Image Captioning. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Vasu Jindal. 2017. A Deep Learning Approach for Arabic Caption Generation Using Roots-Words. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI)*.
- Vasu Jindal. 2018. Generating image captions in Arabic using root-word based recurrent neural networks and deep neural networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 144–151, New Orleans, Louisiana, USA. Association for Computational Linguistics.
- Andrej Karpathy and Li Fei-Fei. 2015. Deep Visual-Semantic Alignments for Generating Image Descriptions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Jungo Kasai, Keisuke Sakaguchi, Lavinia Dunagan, Jacob Morrison, Ronan Le Bras, Yejin Choi, and Noah A. Smith. 2021. Transparent Human Evaluation for Image Captioning. *CoRR*, abs/2111.08940.
- Wuwei Lan, Yang Chen, Wei Xu, and Alan Ritter. 2020. An empirical study of pre-trained transformers for Arabic information extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4727–4734, Online. Association for Computational Linguistics.
- Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. 2020. Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks. In *Computer Vision – ECCV 2020*.
- Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *Computer Vision – ECCV 2014*, pages 740–755, Cham. Springer International Publishing.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. **Bleu: a method for automatic evaluation of machine translation**. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Sabri Monaf Sabri. 2021. Arabic Image Captioning using Deep Learning with Attention. Master’s thesis, University of Georgia.
- Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. **KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media**. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059, Barcelona (online). International Committee for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Conference on Neural Information Processing Systems (NIPS)*.
- Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2014. **CIDEr: Consensus-based Image Description Evaluation**. *CoRR*, abs/1411.5726.
- Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2016. Aggregated Residual Transformations for Deep Neural Networks. *arXiv preprint arXiv:1611.05431*.
- Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *Proceedings of the 32nd International Conference on Machine Learning (PMLR)*.
- Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. 2020. **Multilingual universal sentence encoder for semantic retrieval**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 87–94, Online. Association for Computational Linguistics.
- Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.
- Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. VinVL: Revisiting Visual Representations in Vision-Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5579–5588.
- Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J. Corso, and Jianfeng Gao. 2019. Unified Vision-Language Pre-Training for Image Captioning and VQA. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

A Experiment Hyperparameters

English vs Arabic labels. All experiments were trained and validated with the Flickr8k train- respective val-split. Table 5 shows the exact hyperparameters for the experiments.

Learning curve. All experiments were validated with the Flickr8k val-split and trained on Arabic labels. Table 6 shows the exact hyperparameters for the experiments. Grid search optimization was made on AraBERT and GigaBERT in the interval $\eta \in [1e^{-5}, 7e^{-5}]$ and a step size of $1e^{-5}$.

Large scale. All experiments were validated and tested with the Flickr8k val- respective test-split, and trained on Arabic labels. Table 7 shows the exact hyperparameters for the experiments.

B Complementary Results

Table 8 shows scores for all models trained during the last experiment.

Table 5: Hyperparameters used for the English vs Arabic labels experiments.

Model	Train	Object labels	Learning rate	Batch size	#Epochs
GigaBERT	Flickr8k	eng/ar	1e-4	32	30
mBERT	Flickr8k	eng/ar	1e-4	32	30

Table 6: Hyperparameters and datasets used for the learning curve experiments.

Model	Train	% of dataset	Learning rate	Batch size	#Epochs
AraBERT	Flickr8k	50/75/100	1e-4	32	30
Arabic-BERT	Flickr8k	50/75/100	1e-4	32	30
GigaBERT	Flickr8k	50/75/100	1e-4	32	30

Table 7: Hyperparameters and datasets used for the large scale experiments.

Model	Train	Object labels	Learning rate	Batch size	#Epochs
AraBERT	Flickr8k	ar	3e-5	32	30
	Arabic-COCO	ar	5e-5	32	50
	Arabic-COCO+Flickr8k	ar	3e-5	32	50
	Flickr8k	ar	5e-5	256	30
	Arabic-COCO	ar	9e-5	256	50
	Arabic-COCO+Flickr8k	ar	9e-5	256	50
GigaBERT	Flickr8k	eng	3e-5	32	30
	Arabic-COCO	eng	3e-5	32	50
	Arabic-COCO+Flickr8k	eng	3e-5	32	50
	Flickr8k	eng	9e-5	256	30
	Arabic-COCO	eng	9e-5	256	50
	Arabic-COCO+Flickr8k	eng	9e-5	256	50

Table 8: Our model scores compared to previous models. The highest scores on our test-split are marked in bold. Of all the previous ones, only the model by [ElJundi et al. \(2020\)](#) uses the same test-split as us. Other test-splits are unknown.

Model	Test set	B1	B2	B3	B4	ROUGE-L	METEOR	CIDEr	MUSE	
Jindal (2018)	Flickr8k	0.658	0.559	0.404	0.223	-	0.201	-	-	
Al-muzaini et al. (2018)	COCO & Flickr8k	0.462	0.260	0.190	0.080	-	-	-	-	
Afyouni et al. (2021)	COCO	0.649	0.413	0.241	0.136	0.470	0.408	-	0.78	
ElJundi et al. (2020)	Flickr8k	0.332	0.193	0.105	0.057	-	-	-	-	
AraBERT32-Flickr8k	Flickr8k	0.391	0.246	0.150	0.092	0.331	0.314	0.415	0.671	
AraBERT32-COCO		0.365	0.221	0.129	0.0715	0.31	0.317	0.36	0.669	
AraBERT32-COCO+Flickr8k		0.358	0.216	0.127	0.0715	0.317	0.316	0.364	0.661	
AraBERT256-Flickr8k		0.387	0.244	0.151	0.093	0.334	0.312	0.428	0.668	
AraBERT256-COCO		0.355	0.211	0.122	0.069	0.303	0.313	0.335	0.665	
AraBERT256-COCO+Flickr8k		0.339	0.204	0.12	0.0686	0.302	0.31	0.339	0.655	
GigaBERT32-Flickr8k		0.386	0.241	0.144	0.0827	0.331	0.315	0.403	0.669	
GigaBERT32-COCO		0.36	0.215	0.124	0.0708	0.308	0.311	0.344	0.668	
GigaBERT32-COCO+Flickr8k		0.362	0.216	0.127	0.0675	0.312	0.308	0.359	0.661	
GigaBERT265-Flickr8k		0.376	0.235	0.141	0.0803	0.322	0.313	0.385	0.664	
GigaBERT265-COCO		0.339	0.198	0.113	0.062	0.287	0.306	0.312	0.662	
GigaBERT265-COCO+Flickr8k		0.365	0.217	0.128	0.0705	0.315	0.309	0.373	0.662	
		Δ	0.059 \uparrow	0.053 \uparrow	0.046 \uparrow	0.036 \uparrow				

Plot Writing From *Scratch* Pre-Trained Language Models

Yiping Jin^{1*}, Vishakha Kadam², Dittaya Wanvarie¹

¹Department of Mathematics & Computer Science, Chulalongkorn University, Thailand

²Knorex, 02-129 WeWork Futura, Magarpatta, Hadapsar, Pune, India

Dittaya.W@chula.ac.th

{jinyiping, vishakha.kadam}@knorex.com

Abstract

Pre-trained language models (PLMs) fail to generate long-form narrative text because they do not consider global structure. As a result, the generated texts are often incohesive, repetitive, or lack content. Recent work in story generation reintroduced explicit content planning in the form of prompts, keywords, or semantic frames. Trained on large parallel corpora, these models can generate more logical event sequences and thus more contentful stories. However, these intermediate representations are often not in natural language and cannot be utilized by PLMs without fine-tuning. We propose generating story plots using *off-the-shelf* PLMs while maintaining the benefit of content planning to generate cohesive and contentful stories. Our proposed method, SCRATCHPLOT, first prompts a PLM to compose a content plan. Then, we generate the story's body and ending conditioned on the content plan. Furthermore, we take a generate-and-rank approach by using additional PLMs to rank the generated (story, ending) pairs. We benchmark our method with various baselines and achieved superior results in both human and automatic evaluation ¹.

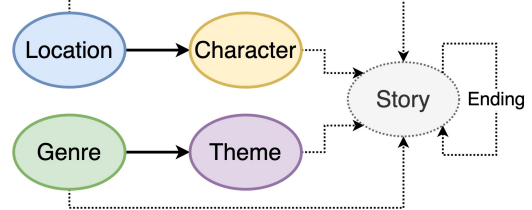
1 Introduction

Automatic story generation aims to produce interesting stories to be read by readers or help writers develop new ideas. However, generating long-form stories is challenging because language models lack global planning (Hua and Wang, 2020; Tan et al., 2021), discourse coherence (Bosselut et al., 2018; Ji and Huang, 2021), and common sense knowledge (Xu et al., 2020; Ji et al., 2020). While individual sentences appear fluent and logical, they do not fit together as a whole and the stories often have no clear content (See et al., 2019; Goldfarb-Tarrant

*Work done while at Knorex.

¹Code and data available at <https://github.com/YipingNUS/scratchplot-story-generation>.

Step 1: Progressive content planning



Step 2: Generate story body

Task: Write a plot summary of a {genre} story featuring {character1} and {character2} in {location} with the main theme "{theme}"
Plot summary: "

Step 3: Generate story ending

Task: Write the ending of a {genre} story.
What happened earlier: {story}
What happens in the end: "

Figure 1: Overview of SCRATCHPLOT. We factorize the elements of a story into four attributes {location, characters, genre, and theme}. We first prompt a PLM to compose them sequentially, then generate the story conditioned on these attributes. When writing the ending of the story, the model additionally conditions on the previously generated story.

et al., 2020). In long text generation, repetitions are also more prevalent, which cause the stories to degrade drastically (Yao et al., 2019).

Interestingly, recent work in long-form story generation relied on *explicit* content planning (Reiter and Dale, 1997), contrary to the prevalent trend of end-to-end learning across NLP tasks. The content plan usually takes the form of prompts (Fan et al., 2018), keywords/keyphrases (Xu et al., 2018; Yao et al., 2019), semantic frames (Fan et al., 2019), or summaries (Sun et al., 2020).

These content plans are usually not in the form of natural language ² and cannot be understood by pre-trained language models (PLMs) without fine-tuning using parallel data. Another subtle problem of modeling story generation as a supervised learn-

²Except for using summaries as the content plan.

ing task is that the model learns common sense and frequently occurring action sequences, like morning routines (Fan et al., 2019). Such an action plan may not be interesting and surprising, which are crucial characteristics of stories.

We propose generating stories using *off-the-shelf PLMs without fine-tuning*. We tap on DINO (Schick and Schütze, 2021), a framework to generate datasets using instructions, to compose stories progressively. Our method, SCRATCHPLOT, is depicted in Figure 1. We firstly prompt a PLM to perform content planning, including the location, characters, genre, and theme. We then generate a story conditioned on these attributes. Finally, we generate story endings and rank them.

The proposed approach yields better stories based on various automatic and human evaluations compared with baselines fine-tuned using parallel data and a PLM with standard prompting. We only experimented with generating English stories. In principle, the approach can be applied to other languages given a reasonable PLM and task descriptions in the target language.

2 Plot Generation From Scratch

DINO (Schick and Schütze, 2021) is a framework to generate labeled NLI datasets (Bowman et al., 2015) using a pre-trained GPT-2 model (Radford et al., 2019). Schick and Schütze (2021) formulated different task descriptions to generate sentence pairs for each category. Instead of generating the sentence pairs at once, they sample the first sentence x_1 , then incorporate it into the task description to sample the second sentence, as shown in Figure 2.

Task: Write two sentences that mean the same thing.
Sentence 1: " x_1 "
Sentence 2: "

Figure 2: Task description to generate a similar sentence by incorporating the first generated sentence, x_1 .

Factorizing Elements of a Story We factorize story generation into multiple stages analogous to generating NLI sentence pairs. We define four main plot elements: location, characters, genre, and theme. These elements are not entirely independent. For example, the genre will influence the theme. We denote these dependencies using solid arrows in Figure 1. We then use different task descriptions to sample these elements sequentially.

Figure 3 shows example task descriptions to generate the genre and theme. We use paraphrases of the task descriptions, and the complete list is presented in Appendix A.

Task: Write a story genre.
Story genre: "

> **action**

Task: Write the twist in a $\langle x_1 \rangle$ story.
Twist: "

> **In the early morning hours on a warm summer's morning, a boy came home and found the family locked in a room.**

Figure 3: Task description for generating genre and theme. $\langle x_1 \rangle$ denotes the generated genre. The example continuations are generated by GPT2-XL.

We apply self-debiasing (Schick et al., 2021) to generate distinct geographical units and male/female names using each task description. Meanwhile, we generate other plot elements without self-debiasing because their task descriptions complement each other. Self-debiasing rewards generated continuations which receive *high* probability conditioned on one task description and *low* probability conditioned on other task descriptions. For example, when generating a male name, we want it to be *unlike* a female name. Specifically, we calculate the token’s probability p_y assigned by the PLM using each task description. The token’s final logit for each task description y is as follows:

$$\delta_y = p_y - \max_{y' \neq y} p_{y'} \quad (1)$$

We sample one value for each plot element except for characters, where we generate a male and a female character. After sampling all plot elements, we fuse them into a single task description to generate the story, as depicted in Step 2 of Figure 1.

Generating Coherent Story Ending Coherent and thoughtful endings are crucial to stories. However, it is not obvious how to write the story ending with PLMs. GPT-2 does not have an $\langle \text{EOS} \rangle$ (end of sequence) token. Therefore, Schick and Schütze (2021) always end the task description with an opening quotation mark (as shown in Figure 2) and generate till the first closing quotation mark. However, the PLM usually generates the first closing quotation mark after a couple of sentences, making it unsuitable for generating long-form stories. Therefore, we generate the story with a fixed length and truncate it till the last complete sentence.

We design a separate task description to write the story ending *explicitly* by providing the story body and asking the PLM to write what happens in the end (Step 3 in Figure 1). As the story ending is usually short, we terminate at the first generated closing quotation mark following Schick and Schütze (2021).

We observe that PLMs sometimes ignore the task descriptions and write generic or irrelevant story endings. Therefore, we propose two methods to rank the story endings. Firstly, we use the next sentence prediction (NSP) task of BERT (Devlin et al., 2019) to measure the coherence between the story and the ending. Specifically, we calculate $\mathcal{P}_{NSP}(b, e)$, where b denotes the story body and e denotes the story ending.

Inspired by previous works in fact-checking with PLMs (Lee et al., 2020, 2021), we use the perplexity score as another metric to measure the story ending’s quality. Specifically, we concatenate the story body and ending to form the input to the PLM: $X = \{x_{b_0}, \dots, x_{b_B}, x_{e_0}, \dots, x_{e_E}\}$, where B and E denote the number of tokens in the story body and ending separately. We then calculate the conditioned perplexity by

$$PPL(X) = \sqrt[E]{\prod_{i=1}^E \frac{1}{p(x_{e_i} | x_{b_0}, \dots, x_{b_B}, \dots, x_{e_{i-1}})}}$$

Note that we use the story body tokens to condition the perplexity, but they do not contribute to the $PPL(X)$.

During inference, we sample multiple (story body, story ending) pairs and use NSP and PPL to rank them ³.

3 Experiments

Experimental Details We use the official implementation of DINO (Schick and Schütze, 2021) ⁴ with the default GPT2-XL language model. We follow the default parameters except setting $k=30$ for top- k sampling and blocking repeating trigrams during generation. For story ending ranking, we use HuggingFace (Wolf et al., 2020) bert-base-uncased checkpoint to calculate the NSP probability and gpt2 (base) to calculate the perplexity.

³NSP the higher, the better. $PPL(X)$ the lower, the better.

⁴<https://github.com/timoschick/dino>

We perform simple post-processing to clean or filter the continuations, such as removing trailing punctuations and filtering continuations that repeat words from the prompt or contain 1st or 2nd person pronouns. The story body must also contain some plot elements to ensure it is contentful and respects the task description. The post-processing is detailed in Appendix B.

We generate plot elements *offline* in batches and store them. When generating stories, we randomly sample each type of plot element and combine them to form a content plan ⁵. Table 1 presents the detailed parameters used for each type of generation.

Element	num	min_len	max_len
Location	20	1	5
Cast	10	1	5
Genre	20	1	5
Theme	10	5	25
Story Body	30	-	100
Story Ending	10	10	50

Table 1: Hyperparameters for generation. For (location, genre, story body), **num** indicates the number of continuations to generate per *task description*. For (cast, theme, story ending), **num** is the number of continuations per *input <X1>* and *task description* combination. Please refer to Appendix A for the number of task descriptions for each plot element.

Baselines We compare with the following three conditional story generation baselines.

- **Fusion** (Fan et al., 2018) ⁶: A seq2seq model with a convolutional encoder and a self-attention decoder generating stories conditioned on a prompt. We use the official checkpoint, which is fine-tuned on the WRITINGPROMPTS dataset with 300k prompt-story pairs.
- **Plan-and-write** (Yao et al., 2019) ⁷: A bidirectional gated recurrent unit (BiGRU) seq2seq model that first predicts the storyline (as specified by a sequence of keywords) from the title. It then generates the story conditioned on both the title and the storyline. We train the model on ROCStories

⁵We sample plot elements following the same sequence in Figure 1 to preserve the dependency among them.

⁶<https://github.com/pytorch/fairseq/tree/main/examples/stories>

⁷<https://bitbucket.org/VioletPeng/language-model>

dataset (Mostafazadeh et al., 2016) for 280 epochs till convergence and follow the default hyper-parameters in the official repository.

- **ProGen** (Tan et al., 2021)⁸: A multi-stage BART seq2seq model (Lewis et al., 2020) using salient keywords as intermediate representations. We use a two-stage seq2seq architecture, where the first seq2seq model takes the input keywords and generates a refined intermediate representation containing keywords with finer-grained details. The second seq2seq model then uses it as input and generates the final story. We fine-tune a BART-base model for both stages using 1k examples randomly sampled from the WRITINGPROMPTS dataset following Tan et al. (2021).

We provide the same *generated* content plans to the baselines to make the comparison fair. We use the generated *theme* as input to the Fusion and Plan-and-write models, which is analogous to the prompt or the title. On the other hand, we extract keywords using TF-IDF following Tan et al. (2021) from all plot elements to prepare the input to ProGen.

We also experiment with a baseline **GPT2-XL without content planning** where we sample a list of stories by providing the instruction “**Task:** Write a plot summary.\n**Plot summary:**”. We limit the story length to 150 tokens in all models for ease of human evaluation⁹.

RQ1: Which story ending ranking performs better for ScratchPlot? We conduct a pair-wise comparison on the following story ending ranking methods: selecting the highest NSP (next sentence prediction) score, the lowest PPL (perplexity) score, and a random story and ending pair. For each pair-wise evaluation, we randomly sample 50 content plans where the two methods select different story endings¹⁰. Each time, we present the annotators two stories in randomized order and ask the annotators to rate which story ends better. Finally, we take the majority vote from three annotators for each comparison and present the result in Table 2. We also show randomly sampled stories and story endings selected by different methods in Table 3.

Based on the human rating, PPL selects more favorable story endings than NSP or Random. Sur-

⁸<https://github.com/tanyuqian/progressive-generation>

⁹Additional experimental details can be found in Appendix C.

¹⁰They may or may not have the same story body.

Method	Win:Lose
NSP vs. Random	19:31
PPL vs. Random*	35:15
PPL vs. NSP *	32:18

Table 2: Pair-wise comparison of the story ending ranking methods. The winning method in each comparison is highlighted in bold. * indicates statistical significance using two-sided Wilcoxon signed-rank test with $p=0.05$.

prisingly, NSP performs worse than random story and ending pairs. We hypothesize it might be due to the weakness of the NSP pre-training task. The negative examples during NSP pre-training are random sentences from the *corpus*, which might be too trivial. Therefore, when the story ending and the story have word overlap, the model often predicts a very high \mathcal{P}_{NSP} close to 1, causing the comparison to be unreliable.

Besides human evaluation on stories generated by PLMs, we also evaluate the story ending ranking methods on the Story Cloze Test dataset (Mostafazadeh et al., 2016). The dataset was created using crowd-sourcing to test models’ commonsense story understanding. Each story contains four preceding sentences, a ‘right ending’ and a ‘wrong ending’. The task is to predict which ending is the right one. Mostafazadeh et al. (2016) instructed the crowdworkers to share at least one of the characters of the story in the ending and to ensure the ending sentence is entirely realistic and sensible when read in isolation. Therefore, the task is non-trivial and shallow techniques barely outperform a random baseline.

We report the Story Cloze Test accuracy in Table 4, along with the three best performing baselines reported in Mostafazadeh et al. (2016). Word2Vec (Mikolov et al., 2013) and Skip-thoughts (Kiros et al., 2015) calculate the semantic representation using the average Word2Vec embedding and Sentence2Vec embedding separately. The models predict the story ending whose embedding is nearest to the preceding story’s embedding. Deep Structured Semantic Model (DSSM) (Huang et al., 2013) employs two jointly trained deep neural networks to project the preceding story context and the story ending into the same semantic space.

NSP and PPL performed comparably with DSSM. Notably, DSSM was trained on the full ROCStories dataset (Mostafazadeh et al., 2016), where the examples of the Story Cloze Test are

Location:	San Francisco
Cast:	John Jones <i>and</i> Evelynn
Genre:	dark fantasy
Theme:	The specter of the future is in the telling
Random:	The protagonist is a woman who has become a cyborg. She must discover who she really is while dealing with the problems of a modern metropolis. John Jones is a mysterious person who claims to be an assassin who is also an inventor and an expert on the cyborgs and other strange and mysterious beings. Evelynn Jones is an attractive and brilliant woman who finds herself caught between the two worlds, and the protagonist’s own conflicted past. [SEP] He had to be the hero! It was a tragedy that he couldn’t be. He was a killer, an assassin. He had to do it and he did it with a smile on his face.
NSP:	As the year comes to a close, two people begin to lose their grip on reality. Evelynn’s father has been dead for several years, but her mother never truly accepted her fate and continues to live a life of denial. John Jones was once a successful businessman, but his fortune was spent by a mysterious corporation that he believed was his own. His life is about to take a turn for the worse as he discovers that Evelynn is living with a life-long secret. [SEP] In order for her to see the future, she’ll have to take the risk.
PPL:	In the past, John Jones used to be a normal person who worked for the government. But after a strange accident, he was taken to a secret facility, where he met the girl he loved, Evelynn, and started a relationship with her. But as the years passed, his memories started to grow more and more vague, and he started to realize that he didn’t really remember how he got into that facility. [SEP] After a few months, John’s memory returned to normal. He and Evelynn had their own children, but the memories remained.

Table 3: Story body and ending selected by different algorithms. We manually insert a [SEP] token to indicate the boundary between the story body and ending.

Method	Accuracy
Word2Vec	0.539
Skip-thought	0.552
DSSM	0.585
NSP	0.580
PPL	0.587

Table 4: The accuracy of various models on the Story Cloze Test test dataset. We copied the results of the first three baselines from Mostafazadeh et al. (2016).

drawn. It also used the Story Cloze Test validation set to perform hyper-parameter tuning. In contrast, NSP and PPL do not require any in-domain data or task-specific training. PPL’s performance was better than NSP, consistent with the previous human evaluation results. Therefore, we use PPL to rank the story endings in subsequent experiments.

RQ2: How does SCRATCHPLOT compare to the baselines? We generate 50 stories using each model and invite three crowdworkers to evaluate

each story on the following fine-grained aspects: naturalness, interestingness, and cohesiveness. We take the average of the scores assigned by the annotators as the final score. Appendix D provides full details of the crowdsourcing evaluation.

Table 5 overviews the result, and Table 8 shows a randomly sampled content plan with stories generated by each model. We notice that the Fusion model tends to generate stories that consist primarily of dialogues, such as the example in Table 8. It is also prone to generating repetitions.

Plan-and-write and ProGen both use a list of keywords as the content plan but are trained on different corpora. Plan-and-write generates short commonsense stories similar to the ROCStories dataset it is trained on. Thanks to its storyline generation, there is a logical sequence among the sentences. However, it lacks diversity in sentence structure. The stories also do not have rich plots and characters, causing them to have the lowest interestingness score among all baselines.

Model	natur	inter	cohes
Fusion	2.13	2.31	1.89
Plan-and-write	2.79	1.70	2.66
ProGen	2.13	3.05	1.88
SCRATCHPLOT	4.04*	3.99*	3.47
w/o content plan	3.64	3.19	3.41

Table 5: Human evaluation result of various models on different aspects. The columns denote naturalness, interestingness, cohesiveness. All scores are on a scale from 1 (worst) to 5 (best). Best scores for each aspect are highlighted in bold. * indicates statistical significance compared with the second best system using two-sided paired t-test with $p=0.01$.

On the other hand, ProGen is trained on the WRITINGPROMPTS dataset, consisting of creative “free-style” stories. Unlike Plan-and-write, the content plan ProGen generates appears like a random bag of words. The stories are also often not logical. The comparison between the two models suggests that while a list of keywords is suitable as the content plan for short stories with mostly single-predicate sentences, it fails to generate cohesive stories with more nuance.

Compared to the baselines, SCRATCHPLOT performed best on all aspects, the improvement in interestingness being especially pronounced.

RQ3: Does unsupervised content planning help? Different from previous work in story content planning, SCRATCHPLOT is entirely unsupervised, i.e., we prompt the same PLM that generates the story to generate the content plan without a need of finetuning.

We first measure the quality of the generated content plan by inviting an *expert* annotator to rate all plot elements generated offline¹¹. We use binary rating (acceptable/unacceptable) for location, cast, and genre. We use a scale from 1 to 5 for theme because it is more subjective. Table 6 presents the average expert rating for each generated plot element. As we can see, the PLM generates simple fields like locations and person names with high quality. However, some generated themes are ambiguous or nonsensical.

SCRATCHPLOT outperformed the baseline without content planning in all aspects in Table 5, demonstrating the contribution of content plans in story generation even when they contain noise.

¹¹We describe the details of plot element generation in Appendix C. Plot elements are much shorter and faster to rate. Therefore, we use an expert annotator for superior accuracy.

Element	Location	Cast	Genre	Theme
Count	43	493	24	117
Score	0.930	0.931	0.792	0.654

Table 6: Expert rated scores for generated plot elements normalized to the range of 0 (worst) to 1 (perfect).

Furthermore, we measure *intra*-story lexical diversity using self-BLEU (Zhu et al., 2018) and *within*-story lexical diversity (or repetition) using *distinct-n* (Li et al., 2016) and summarize the result in Table 7. Unsurprisingly, the baseline without explicit content planning generates less diverse stories because they are sampled by conditioning on the same instruction. It also generates more *within*-story repetitions than SCRATCHPLOT.

Model	self-BLEU		distinct-n	
	$n=1$	$n=2$	$n=1$	$n=2$
Fusion	.805	.483	.079	.316
Plan-and-write	.960	.900	.056	.150
ProGen	.805	.347	.191	.683
SCRATCHPLOT	.782	.353	.228	.705
w/o content plan	.799	.380	.204	.650

Table 7: Result of self-BLEU scores to measure intra-story diversity (the lower the better) and distinct-n scores to measure repetitions (the higher the better). The best results are highlighted in bold.

Qualitatively, Table 8 demonstrates that while the PLM without content planning started the story well, it began “looping” towards the end, a common problem referred to as neural text degeneration (Holtzman et al., 2019). In contrast, SCRATCHPLOT generates a story with natural progression.

Plan-and-write often ignores the input theme completely and predicts common storylines similar to its training data, such as “wanted; decided; bought; started; happy” and “wanted; decided; months; started; felt”. Therefore, it has much worse inter-/intra-story diversity.

4 Related Work

4.1 Story Content Planning

Fan et al. (2018) introduced hierarchical story generation by first generating a prompt then transforming it into a complete story. They proposed a novel fusion-based architecture to improve the relevance between the generation and the input prompt. They also collected a large dataset of 300K (prompt,

Location:	Alameda County
Cast:	John <i>and</i> Cheryl
Genre:	literary fiction
Theme:	The only problem with the house was the people.
Fusion:	“What?” she whispered, as she pulled a small piece from her coat. It was a small piece of paper, with a picture of a young man sitting on that couch. “What?” he said, as he flipped the picture around. “My name is John, and I ’m here to give you a story.” he said, as the photo of his wife came out of his hand and he pulled out a book. The picture was of some kind of book, with the Briggs on it. “What” she said, as she looked at it . “I don’t know,” John said, as he read the book. “It was about to get pretty bad. I don’t like it here.” “It’s not good,” she said. “What did you tell me?” he asked, as the picture was on the couch. “I just can’t believe it
Plan-and-write:	Content plan: <i>wanted; decided; bought; started; happy</i> I wanted to go to the store. I decided to buy a new car. I bought a new car. I started to look for a new car. I was happy to have a new car.
ProGen:	Content plan: <i>object; active; hemperforming; fail; blinked; first; son; joined; wife; bedroom; boys; excited; boy; jerry;</i> The first son I held inside my wife skin and body, I rested my head on her chest and prayed to the waiting Forbiddone, loved by the boy and girl alike, and that the house would bring smiles to all. Jerry Fairey, anxious and excited, pushed his face into my naked flesh. Jerry sounded just like his daddy and chirped, “Grr!” and they laughed together. Jerry took a deep breath, thought of again, “Fuck this. This place is empty, people.” Jerry slept.
SCRATCHPLOT:	John and Cheryl had been divorced for years, and while the divorce was final, there were still legal problems. John had bought a house in a nice part of town, and now, as it was approaching its five-year anniversary, his friends began asking questions about the house. The house was nice, and had nice rooms and a pool and all that. But the people who lived there had some problems. They lived with their mother and her family of five, with no money and no job. On the first day of the week of September, 1995, John and his friends got in their cars and drove back to the house they had been renting in the woods.
SP w/o content plan:	An interesting tale of murder, betrayal, and the power of the human mind. (Summary by Ransom, written by L. Ron Hubbard.) Ransom: “In the year 2392, an entire city has been destroyed and the entire population of a neighboring planet is now on the run. In this story, an ordinary man is caught up in a conspiracy that puts him on the edge of a world-changing discovery.” (Plot summary by L Ron Hubbard, written to a friend.) Plot Summary: A plot summary of one of Hubbard’s books. A summary from Hubbard’s biography, Rise to Power.

Table 8: Stories generated by different models. All models except for “SCRATCHPLOT w/o content plan” use the same plot element(s) listed above. We also show the intermediate content plans generated by Plan-and-write and ProGen.

story) pairs from Reddit’s WRITINGPROMPTS forum. The dataset has been widely used in subsequent work in story generation.

Xu et al. (2018) and Yao et al. (2019) argued that Sequence-to-Sequence (seq2seq) models generate sentences autoregressively and are not good at modeling semantic dependencies *among* sentences. Therefore, they proposed to use a list of keyphrases as the intermediate content plan. Xu et al. (2018) used policy gradient (Sutton et al., 1999) to train the keyphrase extraction module, optimizing towards rewards from the generative module. Yao et al. (2019) explored two strategies: dynamic schema that generates the next keyword in the content plan and the next sentence in the story at each step, and static schema that generates all keywords in the content plan and generates sentences conditioned on the *complete* content plan. They empirically showed that the static schema performed better and conjectured that it generates more coherent stories because it plans the storyline holistically.

Similarly, Tan et al. (2021) used lists of keywords as the content plan. However, their proposed method, ProGen, is a multi-stage Transformers seq2seq model, extracting keywords at different granularities. Each stage takes the output from the previous stage and adds finer-grained details.

Moreover, other representations have been used for story content planning. Fan et al. (2019) and Goldfarb-Tarrant et al. (2020) utilized predicate-argument tuples extracted using Semantic Role Labeling. Sun et al. (2020) employed extractive summarization to generate paragraph summaries from stories as the content plan. Shen et al. (2019) used a hierarchically-structured Variational autoencoders (Bowman et al., 2016) to infer latent representations at word- and sentence-level. During inference, they generate a series of plan vectors before word-level realization.

Unlike previous works, we use *heterogenous* plot elements sampled from a PLM as the content plan (e.g., cast, location, genre). We also do not require any fine-tuning and rely solely on off-the-shelf PLMs.

4.2 Story Ending Generation

Previous work in story ending generation focused mostly on short commonsense stories. Mostafazadeh et al. (2016) introduced ROCStories, a crowd-sourced corpus of 50k five-sentence commonsense stories. The corpus is limited to

non-fictional daily life stories and focuses on being logically meaningful instead of dramatic and entertaining. Mostafazadeh et al. (2016) also introduced the Story Cloze Test task, predicting the correct ending of sample stories from the ROCStories dataset.

Xu et al. (2020) first extracted keywords from the story context in the ROCStories dataset, then retrieved relevant external knowledge from ConceptNet (Speer and Havasi, 2012). Finally, they generated story endings conditioned on the story context and the retrieved knowledge.

Ji et al. (2020) argued that retrieving individual knowledge triples ignores the rich structure within the knowledge graph. To this end, they extracted sub-graphs using the story context and encoded them using a composition-based graph convolutional networks (GCN) (Vashishth et al., 2019). Finally, they performed multi-hop reasoning to generate the story ending.

Rashkin et al. (2020) introduced a simpler approach to story ending generation. Their model, PLOTMACHINES, added special discourse tokens to signal the introduction, body, and conclusion paragraphs in the story. The special token embeddings are trained with the model and help it to learn different writing styles of different parts of the story.

Our story ending generation is most similar to Rashkin et al. (2020) in that we do not perform explicit reasoning but rely on PLMs. However, different from Rashkin et al. (2020), we use natural language instructions instead of trainable embeddings to signal the model to end the story.

Tan et al. (2021) and Sun et al. (2020) used next sentence prediction (NSP) from BERT (Devlin et al., 2019) as an automatic metric to measure intra-sentence coherence. However, we demonstrated in RQ1 that the conditional perplexity score is a more reliable metric. Future work can consider using this metric to measure sentence-level coherence instead.

5 Conclusion

We introduced SCRATCHPLOT, a framework to perform unsupervised content planning for story generation using only pretrained language models (PLM). SCRATCHPLOT achieved strong results compared to supervised baselines fine-tuned on large parallel corpora and a PLM without access to content plans. In future work, we plan to generalize the framework to other types of long-form text.

Acknowledgements

Yiping was supported by the scholarship from ‘The 100th Anniversary Chulalongkorn University Fund for Doctoral Scholarship’ and also ‘The 90th Anniversary Chulalongkorn University Fund (Ratchadaphiseksomphot Endowment Fund)’. In addition, the crowdsourcing human evaluation was funded by Toloka Research Grant ¹². We appreciate their generosity and support for the research community. We would also like to thank the anonymous reviewers for their valuable feedback.

Ethical Considerations

Our proposed method is intended for creative text composition. The generated stories can be either consumed by readers or help writers to come up with new ideas. There are several potential risks if the proposed method is not deployed with care. However, they are inherent from large pre-trained language models (PLMs) instead of intrinsic to our method.

First, PLMs may recall partially from the training data instead of composing stories from scratch. Due to the vast size of the pre-training data, it is not feasible to measure what percentage of the generated stories are “original”. Secondly, the system sometimes generates real person names of famous people as the main characters. It should be noted that the system is for literature purposes and is not meant to be a factual report of real persons or anecdotes. Lastly, the system might generate inappropriate or disrespectful stories to a particular population, such as the genres “biblical epic” and “erotica”. Manual curation or automatic content filtering can be deployed to mitigate this problem.

We relied on crowdworkers to conduct human evaluations in this work. The crowdworkers are from various countries, and the adequate payment differs drastically. Therefore, we target paying \$6.0 per hour. Some of the tasks took longer than we initially estimated, and we issued all crowdworkers a one-time bonus of \$0.2 to compensate.

Although we use a relatively large PLM (GPT2-XL; 1.5 billion parameters), our approach does not require training. Generating a single story takes around 1 minute, consuming 0.003 kWh power based on the max power consumption of the Quadro P5000 we used in the experiment.

¹²<https://toloka.ai/academy/grants>

References

- Steven Bird and Edward Loper. 2004. [NLTK: The natural language toolkit](#). In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.
- Antoine Bosselut, Asli Celikyilmaz, Xiaodong He, Jianfeng Gao, Po-Sen Huang, and Yejin Choi. 2018. [Discourse-aware neural rewards for coherent text generation](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 173–184, New Orleans, Louisiana. Association for Computational Linguistics.
- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. [A large annotated corpus for learning natural language inference](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.
- Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. [Generating sentences from a continuous space](#). In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2018. [Hierarchical neural story generation](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2019. [Strategies for structuring story generation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2650–2660, Florence, Italy. Association for Computational Linguistics.
- Seraphina Goldfarb-Tarrant, Tuhin Chakrabarty, Ralph Weischedel, and Nanyun Peng. 2020. [Content planning for neural story generation with aristotelian rescoring](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4319–4338, Online. Association for Computational Linguistics.

- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *Proceedings of the Seventh International Conference on Learning Representations*, New Orleans, USA.
- Xinyu Hua and Lu Wang. 2020. [PAIR: Planning and iterative refinement in pre-trained transformers for long text generation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 781–793, Online. Association for Computational Linguistics.
- Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338, San Francisco, USA.
- Haozhe Ji and Minlie Huang. 2021. [DiscoDVT: Generating long text with discourse-aware discrete variational transformer](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4208–4224, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Haozhe Ji, Pei Ke, Shaohan Huang, Furu Wei, Xiaoyan Zhu, and Minlie Huang. 2020. [Language generation with multi-hop reasoning on commonsense knowledge graph](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 725–736, Online. Association for Computational Linguistics.
- Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. *Advances in neural information processing systems*, 28.
- Nayeon Lee, Yejin Bang, Andrea Madotto, and Pascale Fung. 2021. [Towards few-shot fact-checking via perplexity](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1971–1981, Online. Association for Computational Linguistics.
- Nayeon Lee, Belinda Z. Li, Sinong Wang, Wen-tau Yih, Hao Ma, and Madian Khabsa. 2020. [Language models as fact checkers?](#) In *Proceedings of the Third Workshop on Fact Extraction and VERification (FEVER)*, pages 36–41, Online. Association for Computational Linguistics.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. [A diversity-promoting objective function for neural conversation models](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.
- Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. [A corpus and cloze evaluation for deeper understanding of commonsense stories](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Hannah Rashkin, Asli Celikyilmaz, Yejin Choi, and Jianfeng Gao. 2020. [PlotMachines: Outline-conditioned generation with dynamic plot state tracking](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4274–4295, Online. Association for Computational Linguistics.
- Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87.
- Timo Schick and Hinrich Schütze. 2021. [Generating datasets with pretrained language models](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6943–6951, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Timo Schick, Sahana Udupa, and Hinrich Schütze. 2021. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in nlp. *arXiv preprint arXiv:2103.00453*.
- Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D. Manning. 2019. [Do massively pretrained language models make better storytellers?](#) In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 843–861, Hong Kong, China. Association for Computational Linguistics.
- Dinghan Shen, Asli Celikyilmaz, Yizhe Zhang, Liqun Chen, Xin Wang, Jianfeng Gao, and Lawrence Carin.

2019. Towards generating long and coherent text with multi-level latent variable models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2079–2089, Florence, Italy. Association for Computational Linguistics.
- Robyn Speer and Catherine Havasi. 2012. Representing general relational knowledge in ConceptNet 5. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12)*, pages 3679–3686, Istanbul, Turkey. European Language Resources Association (ELRA).
- Xiaofei Sun, Chun Fan, Zijun Sun, Yuxian Meng, Fei Wu, and Jiwei Li. 2020. Summarize, outline, and elaborate: Long-text generation via hierarchical supervision from extractive summaries. *arXiv preprint arXiv:2010.07074*.
- Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.
- Bowen Tan, Zichao Yang, Maruan Al-Shedivat, Eric Xing, and Zhiting Hu. 2021. Progressive generation of long text with pretrained language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4313–4324, Online. Association for Computational Linguistics.
- Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2019. Composition-based multi-relational graph convolutional networks. In *International Conference on Learning Representations*, New Orleans, USA.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Jingjing Xu, Xuancheng Ren, Yi Zhang, Qi Zeng, Xiaoyan Cai, and Xu Sun. 2018. A skeleton-based model for promoting coherence among sentences in narrative story generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4306–4315, Brussels, Belgium. Association for Computational Linguistics.
- Peng Xu, Mostofa Patwary, Mohammad Shoeybi, Raul Puri, Pascale Fung, Anima Anandkumar, and Bryan Catanzaro. 2020. MEGATRON-CNTRL: Controllable story generation with external knowledge using large-scale language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2831–2845, Online. Association for Computational Linguistics.
- Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7378–7385, Honolulu, USA.
- Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Tegygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100.

A Full List of Task Descriptions

Table 9 shows the complete list of task descriptions to generate various plot elements for content planning.

B Post-Processing

We perform various post-processing depending on the plot elements. We rely on simple heuristics based on common errors we observe.

Including tailing punctuations For some plot elements, we expect a phrase instead of a whole sentence. However, the PLM sometimes inserts a punctuation mark, such as a full stop or a comma. Therefore, we recursively remove punctuations at the end till the last character is a letter.

Repeating the prompt We observe that the PLM sometimes repeats or rephrases the task description instead of trying to perform the task. Therefore, we filter out continuations that contain *any* word in the task description (excluding stop words and the text replacing the placeholder <X1>).

Generating 1st or 2nd person pronouns We usually do not expect first or second-person pronouns in a story plot. When the model generates plot elements containing first or second-person pronouns, it is often generic or opinionated, such as “I’ll try not to think about it” or “You will not fail me.” Therefore, we filter continuations containing a first or second-person pronoun of any case.

Ignoring task description When generating the story, we want to ensure that it includes essential plot elements specified in the task description. Therefore, we filter out a story if it contains fewer

Element	Task Description
Location	Task: Write the name of a country.\n Country: “ Task: Write the name of a province.\n Province: “ Task: Write the name of a city.\n City: “ Task: Write the name of a county.\n County: “
Cast	Task: Write the male character’s full name in a story that happened in <X1>. Full name: “ Task: Write the female character’s full name in a story that happened in <X1>. Full name: “
Genre	Task: Write a story genre.\n Story genre: “ Task: Write a literary genre.\n Literary genre: “ Task: Write a novel genre.\n Novel genre: “
Theme	Task: Write the main point from a <X1> story.\n Main point: “ Task: Write the twist in a <X1> story.\n Twist: “ Task: Write the lesson learned from a <X1> story.\n Lesson learned: “ Task: Write the spectacle of a <X1> story.\n Spectacle: “

Table 9: Full list of task descriptions to generate each element. <X1> denotes the previously generated element. Story body and story ending generation both use a single task description as shown in Figure 1.

than two of the following {male character’s first name, female character’s first name, location}.

Table 10 overviews the post-processing applied when generating each type of output.

C Additional Experimental Setups

During training/generation, we use the tokenizers associated with the corresponding PLM in the HuggingFace library (Wolf et al., 2020). When calculating diversity and repetition, we use NLTK (Bird and Loper, 2004) to perform word tokenization. We calculate self-BLEU scores using NLTK’s `sentence_bleu` method by treating each example as the reference in each round and averaging the BLEU scores over the whole dataset.

All experiments in this work are conducted on cloud instances with an NVIDIA Quadro P5000 GPU (16GB vRAM). The time to generate a story is roughly 1 minute, which includes generating multiple story bodies and endings and using scoring models to select the best candidate. Since we do not require any fine-tuning, using a CPU to perform inference is also possible. The reader can consider using a smaller GPT2-medium PLM instead of GPT2-XL when the resource is limited. The generation quality is comparable based on our observation.

D Details of Crowdsourcing Evaluation

We conducted the crowdsourcing evaluations on the Toloka platform¹³. In this section, we detail the specification of the annotation tasks, the quality control measures, and the stats of the annotation.

D.1 Annotation Task Specifications

For the fine-grained evaluation, we decompose it into a separate annotation task per aspect so that the annotators can focus on evaluating a single aspect and avoid context switching.

Fine-grained evaluation Rate each story in the following aspects on a scale of 1 (worst) to 5 (best).

- **Naturalness:** Is the story fluent and understandable? The language should be natural. Minor grammatical errors are acceptable if they do not affect understanding the story.
- **Interestingness:** Is the story interesting to readers? Rate this aspect as objective as possible. Assuming someone familiar with the particular genre, will the story interest them?
- **Cohesiveness:** Is the story cohesive and logical? Common problems include mixing up the characters and introducing illogical event sequences (unless it appears like a deliberate choice).

¹³<https://toloka.ai/>

Post-processing	Location	Cast	Genre	Theme	Body	Ending
Remove trailing punctuations	✓	✓	✓			
Filter repeating prompt	✓	✓	✓	✓	✓	✓
Filter 1st & 2nd person pronouns				✓	✓	✓
Filter by plot elements					✓	

Table 10: Post-processing steps applied for each generation task.

Figure 5 shows the annotation interface and Figure 6, 7, 8 shows the detailed annotation instructions.

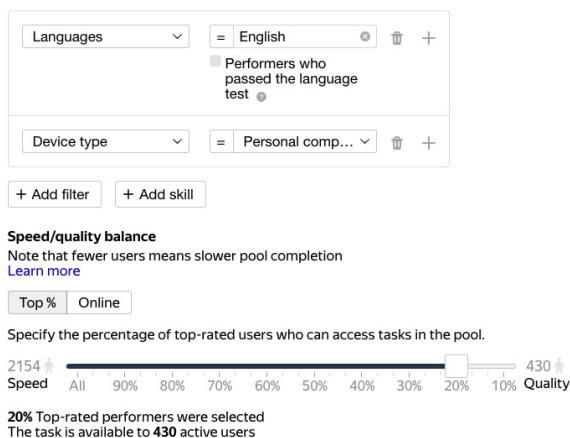


Figure 4: Audience filter for the annotation task pool.

Story ending evaluation Indicate which of the two stories has a better ending. A good story ending should be relevant to the story, logical, conclusive, and thoughtful. Figure 9 shows the full annotation instructions and Figure 10 shows the annotation UI.

D.2 Quality Control

We select crowdworkers who are fluent in English and among the 20% top-rated performers. Figure 4 shows a screenshot of the annotator filter. Additionally, they have to pass a short training session and correctly answer 3 out of 4 training questions to be selected for the main evaluation.

During annotation, we apply various quality control rules, including limiting each annotator to no more than 50 tasks, adding occasional captcha to block bots, banning users who consistently submit tasks too fast (less than 5 seconds for fine-grained evaluation and less than 10 seconds for story ending evaluation), and banning users who skip more than 5 tasks in a row.

D.3 Annotation Task Stats

We paid \$0.05 for each fine-grained evaluation task. On average, it took around 30 seconds to complete each task, making the average earning \$6 an hour. There are around 40 crowdworkers evaluating for each aspect. Figure 11 shows an example pool stats for the naturalness evaluation.

We paid \$0.1 for each story ending evaluation task, which takes on average 1 minute 13 seconds to complete. There are in total 20 crowdworkers participating in this evaluation task.

The overall budget we spent on all crowdsource evaluations is \$300 (including payment and bonus to crowdworkers and platform fees).

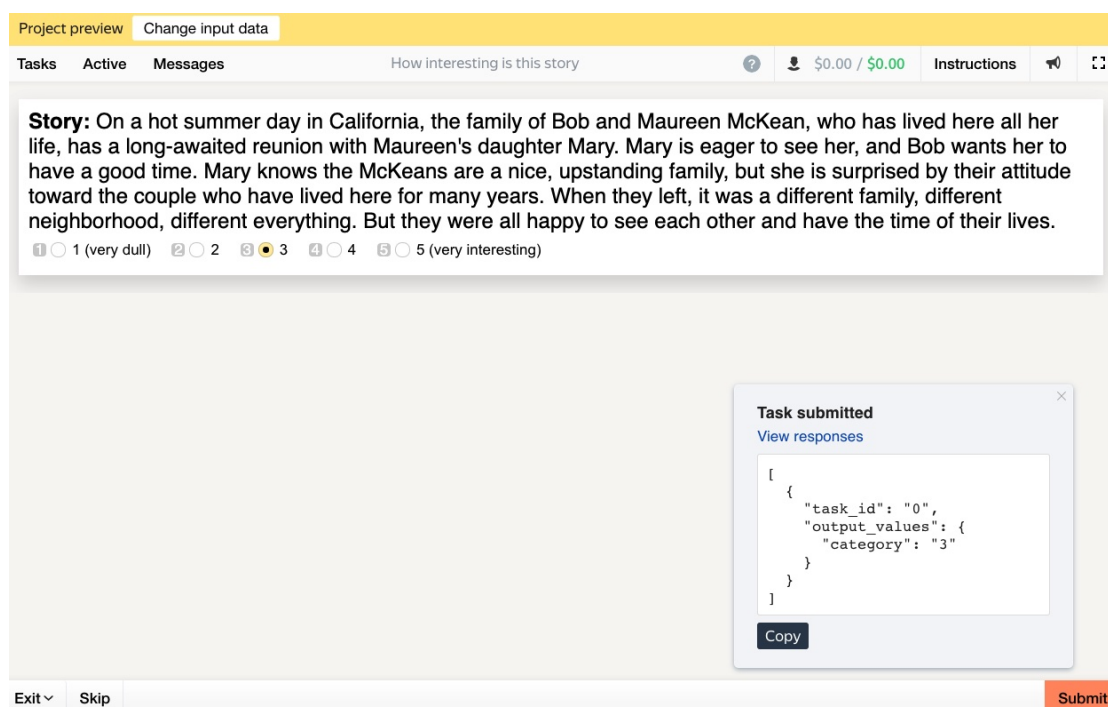


Figure 5: Annotation interface for the interestingness aspect. The interface for other aspects are analogous and we omit them for brevity.

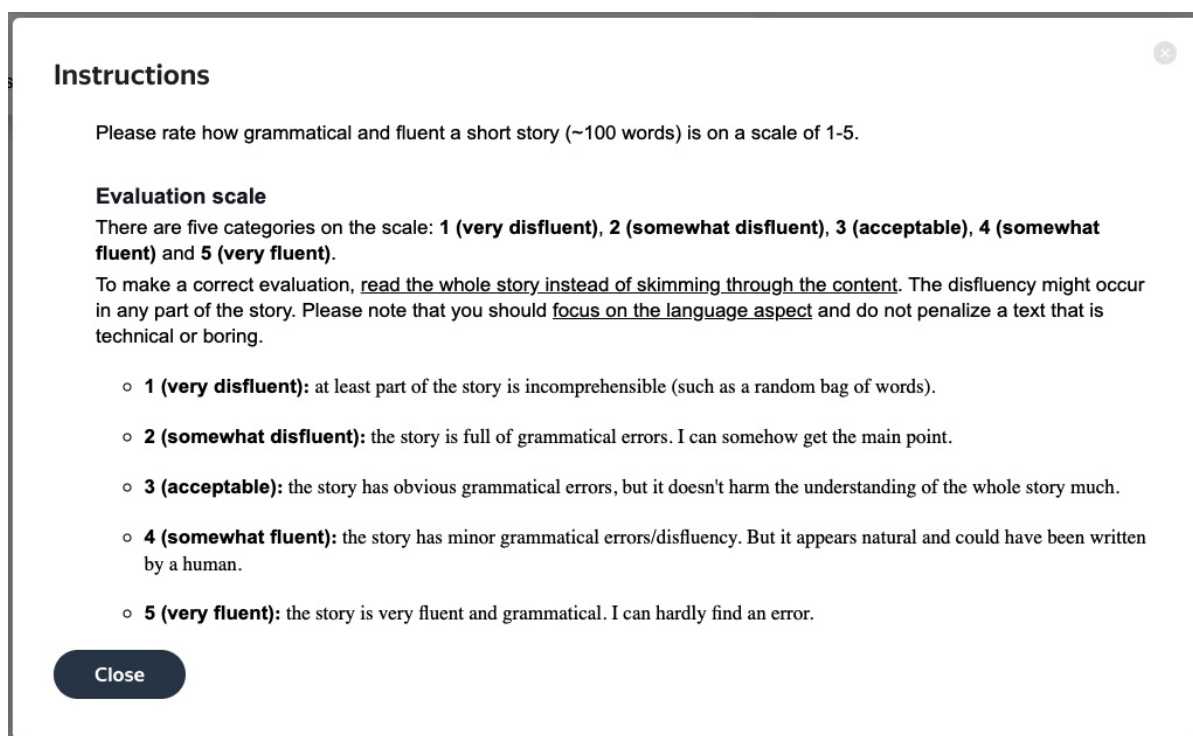


Figure 6: Annotation instructions for the naturalness aspect.

Instructions

Please rate how interesting a short story (~100 words) is on a scale of 1-5.

Evaluation scale

There are five categories on the scale: **1 (very dull)**, **2 (somewhat dull)**, **3 (acceptable)**, **4 (somewhat interesting)** and **5 (very interesting)**.

To make a correct evaluation, read the whole story instead of skimming through the content. Ignore minor grammatical errors and focus on the overall quality.

- **1 (very dull)**: the story is by no means interesting. It's a waste of time to read it. If the text isn't understandable or isn't even a story, you should also assign the score 1.
- **2 (somewhat dull)**: the story is boring. It would fail to interest the majority of the readers.
- **3 (acceptable)**: the story has some twists. However, overall it's not there yet.
- **4 (somewhat interesting)**: the story has an interesting or unexpected plot. It may interest some audiences.
- **5 (very interesting)**: the story is very interesting. People would love to read it.

Close

Figure 7: Annotation instructions for the interestingness aspect.

Instructions

Please rate how coherent a short story (~100 words) is on a scale of 1-5.

Evaluation scale

There are five categories on the scale: **1 (very incoherent)**, **2 (somewhat incoherent)**, **3 (acceptable)**, **4 (somewhat coherent)** and **5 (very coherent)**.

To make a correct evaluation, read the whole story instead of skimming through the content. The coherence problem usually appears across multiple sentences. While individual sentences may seem meaningful, sometimes, they don't fit together as a whole story.

- **1 (very incoherent)**: the story doesn't make sense. It doesn't have clearly defined characters and a plot.
- **2 (somewhat incoherent)**: it's difficult to understand the main plot. Some events seem out of order.
- **3 (acceptable)**: the story has obvious logical errors, but I can understand what it's about.
- **4 (somewhat coherent)**: the story plot is more or less clear. However, there are minor logical errors.
- **5 (very coherent)**: the story is very coherent. There's a clear progression of events. It could have been written by a human.

Close

Figure 8: Annotation instructions for the cohesiveness aspect.

Instructions

Please rate which of the two stories has a better ending. While the story endings are highlighted in **bold** for your convenience, please read the whole story instead of only the endings. A good story ending should be:

- **Relevant:** relevant and coherent with the story. If it seems irrelevant, it's a poor story ending.
- **Logical:** the story ending should be logically derived from the story. While surprises might occur in stories, it should be intentional and understandable.
- **Conclusive:** ideally, story endings should conclude the story and don't leave any open loops. However, certain genres of stories tend to leave an open question to the reader. Please perform your judgement on whether a story ending is appropriate.
- **Thoughtful:** ideally, a story ending should convey a message or insight besides continuing the narrative.

Additional Notes:

- "better" doesn't imply it must be a "happy ending". A tragic ending, if appropriate should be rated better than a low-quality happy ending.
- While the qualities mentioned above are ideal, they may not satisfy in every story. Please contrast the two stories and pick the one that has a relatively better ending.
- Sometimes, the story bodies contain minor errors. While you should read them to get the context, the evaluation should be focused on the story ending.

Close

Figure 9: Annotation instructions for the story ending evaluation.

Project preview Change input data

Tasks Active Messages Which story ending is better \$0.00 / \$0.00 Instructions

Story A

A group of living toys, who assume lifelessness around humans, are preparing to move into a new house with their owner Andy Davis, his sister Molly and their single mother. The toys become uneasy when Andy has his birthday party a week early; to calm them, Sheriff Woody, Andy's favorite toy and their leader, sends Sarge and his green army men to spy on the gift opening with a baby monitor. The other toys are relieved when Andy receives nothing that could replace them. Andy then receives a last-minute surprise gift – a Buzz Lightyear action figure who believes he is a real space ranger. **Buzz impresses the other toys with his various features and becomes Andy's new favorite, making Woody jealous.**

Story B

A group of living toys, who assume lifelessness around humans, are preparing to move into a new house with their owner Andy Davis, his sister Molly and their single mother. The toys become uneasy when Andy has his birthday party a week early; to calm them, Sheriff Woody, Andy's favorite toy and their leader, sends Sarge and his green army men to spy on the gift opening with a baby monitor. The other toys are relieved when Andy receives nothing that could replace them. Andy then receives a last-minute surprise gift – a Buzz Lightyear action figure who believes he is a real space ranger. **On his first day of school, Forrest meets a girl named Jenny Curran, and the two become best friends.**

Story A's ending is better Story B's ending is better

Exit Skip Submit

Figure 10: Annotation interface for the pair-wise story ending evaluation.

POOL STATISTICS

36 sec Average assignment submit time	4 sec Approximate finish time	30.00 (+ 9.00) \$ Budget spent (+ fee)	30.00 (+ 9.00) \$ Approximate budget (+ fee)
255 people ↑ Active users with access to pool	37 people ↑ Interested in pool	37 people ↑ Submitted in pool	16.22 📄 Submitted assignments per performer
		1 items 📄 Expired task suites	0 items 📄 Skipped task suites

Figure 11: The annotation pool stats for the naturalness evaluation.

Paraphrasing via Ranking Many Candidates

Joosung Lee

Kakao Enterprise Corp., South Korea

rung.joo@kakaenterprise.com

Abstract

We present a simple and effective way to generate a variety of paraphrases and find a good quality paraphrase among them. As in previous studies, it is difficult to ensure that one generation method always generates the best paraphrase in various domains. Therefore, we focus on finding the best candidate from multiple candidates, rather than assuming that there is only one combination of generative models and decoding options. Our approach shows that it is easy to apply in various domains and has sufficiently good performance compared to previous methods. In addition, our approach can be used for data augmentation that extends the downstream corpus, showing that it can help improve performance in English and Korean datasets.

1 Introduction

Paraphrasing is the task of reconstructing sentences with different words and phrases while maintaining semantic meaning when a source sentence is given. The paraphrase system can be used to add variability to a source sentence and expand it to sentences containing more linguistic information. Paraphrasing has been studied and closely associated with various NLP tasks such as data augmentation, information retrieval, and question answering.

The supervised approach (Patro et al., 2018) to paraphrase is that the model can be trained to generate the paraphrase directly, but requires a parallel dataset. These parallel datasets are expensive to create and difficult to cover various domains. Therefore, in recent years, many studies (Bowman et al., 2016; Miao et al., 2019; Liu et al., 2020a) have been conducted on an unsupervised approach to learning paraphrase generation using only the corpus. In addition, there are studies (Mallinson et al., 2017; Thompson and Post, 2020) that attempt to paraphrase with machine translation learned with a translation corpus (e.g., language pairs shown

in WMT¹) that has been released widely publicly. Various models have been developed in these methods, but only one model cannot guarantee the best performance for all datasets. Therefore, our goal is not to focus on designing language models or machine translation, but to find best candidates among paraphrases generated by various methods and use them for downstream tasks.

We paraphrase based on a machine translation that can vectorizes sentences with the same meaning in different languages into the same latent representation through an encoder. Our system paraphrases the source sentences with two frameworks and several decoding options and is described in Section 2. Paraphrase candidates generated in various combinations are ranked according to fluency, diversity, and semantic score. Finally, the system selects a paraphrase that has different words from the source sentence, but is naturally and semantically similar.

The performance and effectiveness of the proposed system are verified in two ways. First, our model is evaluated against a dataset provided with a paraphrase pair. We use QQP (Quora Question Pairs) (Patro et al., 2018) and Medical domain dataset (McCreery et al., 2020) and are evaluated by multiple metrics by comparing generated paraphrase and gold reference. The second is to use our system as data augmentation in downstream tasks. We augment financial phrasebank (Malo et al., 2014) and hate speech (eng) (de Gibert et al., 2018) in English and hate speech (kor) (Moon et al., 2020) in Korean to improve the performance of the classification task.

Our system outperforms the previous supervised and unsupervised approaches in terms of the semantic, fluency, and diversity scores shows similar performance to the latest unsupervised approaches. In addition, our system shows performance improvement of downstream tasks, which is a sce-

¹<http://www.statmt.org/wmt20/>

nario where training data is limited. Finally, our paraphrase has the advantage that it can be applied not only to English but also to various languages.

2 Methods

2.1 Pre-trained Model

We use M2M100 (Fan et al., 2020) as backbone models so that it can be used not only in English but also in various languages. M2M100 is a multilingual encoder-decoder model that can handle 100 languages, where M2M100-small and M2M100-large two versions are used.

2.2 Generate Paraphrase Candidates

We generate paraphrase candidates as follows with two methods according to the combination of encoder and decoder.

2.2.1 Src-Encoder+Src-Decoder

The first framework-1 is to use only one language (i.e. source language). Thus, the decoder generates paraphrase candidates directly from the encoded vector of the source sentence. This framework is similar to auto-encoder, but since the paraphrase model is based on a translation system, it has the purpose of generating the same meaning rather than reconstruction.

2.2.2 Round-trip Translation

If a candidate sentence is generated with only Section 2.2.1, the diversity decreases, so the second framework-2 uses two languages to generate more candidates. In other words, we use the round-trip translation mentioned in the Sennrich et al. (2016) to translate the source sentence into the target sentence and translate it back into the source sentence. Because back-translation depends on the performance of the translation system, context information can sometimes be lost, but it can generate various candidates. M2M100 supports 100 languages, but we selected and used English, Korean, French, Japanese, Chinese, German, and Spanish as the language pool.

2.2.3 Decoder Options

When generating paraphrase candidates, we expand the set of candidates by adding various options to the decoder.

In the framework-1, beam search with the beam size of 10 is used and the top-5 candidate sentences are generated. In addition, the following blocking

restrictions are additionally applied. (1) Output tokens are restricted so that they do not overlap more than half of the length of the source sentence in succession with the source tokens. (2) It is prevented from generating repetitive 3-grams within the output sentence.

In the framework-2, 3-beam-search is used in both the forward and backward paths, and the top-1 candidate sentence is generated, and the rest are the same as the framework-1.

2.3 Ranking and Filtering

We filter through various scores to select the best paraphrase among paraphrase candidates. All ranking and filtering processes measure the score in all lowercase letters to eliminate differences due to uppercase and lowercase letters. The candidates with poor scores in each filtering step are discarded.

2.3.1 Overlapping

We remove the overlapping sentences among the candidates that are different from the source sentence. Even in different sentences, candidates that differ only in spaces or by substitution of upper and lower case letters are considered to be the same sentence. The remaining sentences that have been filtered in this section are called *overlap_cands*.

2.3.2 Diversity

We measure diversity by comparing *overlap_cands* and source sentences. We use word error rate (Morris et al., 2004) as diversity metrics, where the higher the score, the higher the diversity. WER (word error rate) refers to the Levenshtein distance between the source sentence and the candidates, and works at the word level instead of the phoneme level. Originally, WER was proposed to measure the performance of an automatic speech recognition system, but we use it to measure the difference between sentences. In this step, only $\min(5, \text{num}(\textit{overlap_cands})/2)$ sentences with a high diversity score are left, and this is called *diversity_cands*.

2.3.3 Fluency

To evaluate fluency, we measure PPL (perplexity) using a language model. Fluency indicates the naturalness of the sentence, and the lower the PPL, the better the fluency. We use GPT2-medium (Radford et al., 2019) as the language model and leave only $\min(3, \text{num}(\textit{diversity_cands})/2)$ sentences with a low PPL, and call this *fluency_cands*.

Dataset	train	dev	test
Financial Phrasebank	1834	203	227
Hate Speech (eng)	1081	220	255
Hate Speech (kor)	1421	789	471

Table 1: Downstream Datasets

2.3.4 Semantic

Semantic score measures using a bidirectional pre-trained language model. BERTScore (Zhang* et al., 2020) leverages the contextual embeddings and matches words in the candidates and the source sentence by cosine similarity. Higher scores mean semantic similarity, and we use RoBERTa-large (Liu et al., 2020b) in BERTScore. We measure the semantic score using the source sentence as a reference and *fluency_cands* as candidates.

2.4 Details

If the source sentence is very short or given a simple structure, in order to obtain more candidates, the decoder options in Section 2.2.3 are restricted so that the source and output sentences do not overlap more than 2-grams.

3 Experiments

Our training and tests are tested on a single V100 GPU, and the details are described in this Section.

3.1 Paraphrasing

3.1.1 Dataset

To measure the performance of paraphrase systems, we used Quora Question Pairs (QQP) test data with 30,000 pairs used in Patro et al. (2018) and medical domain dataset (McCreery et al., 2020).

3.1.2 Evaluation Metrics

We measure the semantic, diversity, and fluency scores of paraphrases. To set Section 2.3 and the evaluation metric differently, diversity uses Isacrebleu (inverser-sacrebleu). Isacrebleu is calculated as 100-sacrebleu (Post, 2018), and the higher the number of overlapping n-grams between candidates and source sentences, the lower the score. The semantic score is measured by comparing it with the gold references provided by the dataset and using Bleurt (Sellam et al., 2020). Bleurt is an evaluation metric trained on biased training data so that BERT can model human judgments. We use bleurt-base-128 as the model for Bleurt. When measuring Fluency, GPT2-small is used as a language model.

3.2 Downstream Task

To demonstrate the usefulness of our approach, we paraphrase several downstream datasets to experiment with the effects of data augmentation. We test sentence classification in the domains of financial phrasebank (Malo et al., 2014) and hate speech (de Gibert et al., 2018) to check usefulness in various domains. It is also paraphrased in hate speech (Moon et al., 2020) in Korean to check its usefulness not only in English but also in other languages.

We download the datasets using huggingface’s dataset library ². Financial phrasebank and hate speech (eng) are randomly divided into training, validation, and test data because only training data is provided. Hate speech (kor) provides training and test data, so a portion of the training data is used as validation. Since our purpose is to confirm the performance improvement with data augmented by paraphrase in a scenario where there is insufficient data, we preprocess hate speech as follows. (1) In hate speech (eng), the data class is unbalanced, so the data of the class that appears excessively is discarded at random to balance the data. Also, since the amount of existing training data is sufficiently large, in order to limit it to a scenario where data is insufficient, we only use 50% of the randomly balanced training data. (2) Hate speech (kor) similarly has enough training data, so only 20% of the training data is randomly used for training. Table 1 shows the statistics of the processed downstream tasks and the performance is measured by accuracy.

4 Results

4.1 Paraphrasing

Table 2 shows the performance of paraphrase. **Edlp** and **Edlps** are supervised learning models introduced in Patro et al. (2018), ED, L, P and S stand for encoder-decoder, cross-entropy, pair-wise discriminator loss, and parameter sharing, respectively. **CGMH** (Miao et al., 2019) uses Metropolis-Hastings sampling in word space to generate constrained sentences. **UPSA** (Liu et al., 2020a) is a method of generating Unsupervised Paraphrase through Simulated Annealing, which searches the sentence space towards this objective by performing a sequence of local edits. **M2M100** is an M2M-large model that paraphrases source sentences with greedy search (top-1) in framework-1.

²https://huggingface.co/datasets/{financial_phrasebank,hate_speech18,kor_hate}

Methods		QQP			Medical		
		Semantic	Diversity	Fluency	Semantic	Diversity	Fluency
		Bleurt	isacrebleu	PPL	Bleurt	isacrebleu	PPL
supervised	Edlp	-1.066	86.843	585.384	-	-	-
	Edlps	-0.857	83.504	597.024	-	-	-
unsupervised	UPSA	-0.729	65.749	392.833	-1.351	89.418	476.069
	CGMH(50)	-0.842	65.35	556.163	-1.405	88.95	818.307
	M2M100	0.036	43.539	346.17	-0.561	35.688	296.672
	Ours	0.083	69.421	171.61	-0.508	68.735	158.76
source	input sentence	0.124	0	270.781	-0.523	0	249.107
	gold reference	1	72.002	278.163	1	88.632	171.786

Table 2: Paraphrasing performance of our approach and previous studies in QQP and Medical. The parentheses of CGMH mean iteration in which the sentence is modified with sample time. Bold text means the best performance.

Our approach achieves the best performance in terms of semantic and fluency scores than previous studies of supervised and unsupervised methods. The diversity score is not the best performance, but it achieves a score comparable to other models. M2M100, which generates a paraphrase using the same model as ours, achieves the second semantic score, but the diversity is worse than the previous methods. That is, the method of generating simply as a translation model as one option is not perfect, and the rate of generating by copying source sentences from M2M100 in the QQP dataset is 8.41%.

4.2 Downstream Task

Table 3 shows the performance of sentence classification, which are downstream tasks. BERT-base is a bidirectional pre-trained language model. Transformer has the same architecture, but trains from scratch. Both models are trained five times and are the average of the measured performances. We observe that the performance of models is improved when the augmented corpus is used for training.

Because BERT is a pre-trained language model trained from numerous corpuses, it has the ability to extract contextual knowledge. Nevertheless, adding the corpus augmented with paraphrase improves the performance, which shows that it helps training even when fine-tuning the pre-trained language model. Transformers trained from scratch do not have general knowledge of the language, so performance changes through data augmentation are large. Performance is greatly improved in financial and hate speech (eng), but data augmentation in Transformer degrades performance in hate speech (kor). We find that Transformer can learn rich representations through paraphrasing of training data, but performance degradation can occur on fixed test data with a small amount of data.

Data augmentation through M2M also shows a similar pattern to ours, but the performance im-

Methods	augmentation	Financial	Hate Speech (eng)	Hate Speech (kor)
BERT-base	x	95.3	64.94	52.78
	M2M	95.15	66.2	54.52
	Ours	96.33	68.31	55.03
Transformer	x	80.47	53.24	52.27
	M2M	85.9	55.69	49.26
	Ours	86.49	63.14	51.04

Table 3: Accuracy of fine-tuned models in downstream tasks. The performance of each model is the average of the values measured by experimenting five times.

provement is small and the performance degradation is large. We infer that, as shown in Section 4.1, the paraphrase performance difference and M2M generate some overlapping sentences.

5 Conclusion

We propose a system that generates various paraphrase candidates and finds the best candidate through multiple scores, which avoids the risk of relying on one model and one decoding option. Our approach captures semantic information better than the previous supervised and unsupervised methods and generates more natural sentences. The diversity score also achieves similar performance to the state-of-the-art unsupervised method. However, our approach may suffer from speed issues for inferring heavy models in parallel on one server. For actual paraphrase use, it will be effective to extract candidates along with a simple model such as n-gram.

Our system shows that when data is insufficient in various domains, the classification performance can be improved through data augmentation through our paraphrasing. Our approach is easily extensible across many domains and languages, and we hope to help with a variety of NLP tasks, such as classification tasks with little data.

References

- Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. [Generating sentences from a continuous space](#). In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.
- Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. 2018. [Hate Speech Dataset from a White Supremacy Forum](#). In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, Brussels, Belgium. Association for Computational Linguistics.
- Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2020. Beyond english-centric multilingual machine translation. *arXiv preprint arXiv:2010.11125*.
- Xianggen Liu, Lili Mou, Fandong Meng, Hao Zhou, Jie Zhou, and Sen Song. 2020a. [Unsupervised paraphrasing by simulated annealing](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 302–312, Online. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. [Ro{bert}a: A robustly optimized {bert} pretraining approach](#).
- Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2017. [Paraphrasing revisited with neural machine translation](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 881–893, Valencia, Spain. Association for Computational Linguistics.
- P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65.
- Clara H. McCreery, Namit Katariya, Anitha Kannan, Manish Chablani, and Xavier Amatriain. 2020. [Effective transfer learning for identifying similar questions: Matching user questions to covid-19 faqs](#). In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '20*, page 3458–3465, New York, NY, USA. Association for Computing Machinery.
- Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. 2019. [Cgmh: Constrained sentence generation by metropolis-hastings sampling](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6834–6842.
- Jihyung Moon, Won Ik Cho, and Junbum Lee. 2020. [BEEP! Korean corpus of online news comments for toxic speech detection](#). In *Proceedings of the Eighth International Workshop on Natural Language Processing for Social Media*, pages 25–31, Online. Association for Computational Linguistics.
- A. Morris, V. Maier, and P. Green. 2004. From wer and ril to mer and wil: improved evaluation measures for connected speech recognition. In *INTERSPEECH*.
- Badri Narayana Patro, Vinod Kumar Kurmi, Sandeep Kumar, and Vinay Namboodiri. 2018. [Learning semantic sentence embeddings using sequential pairwise discriminator](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2715–2729, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Matt Post. 2018. [A call for clarity in reporting BLEU scores](#). In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. [BLEURT: Learning robust metrics for text generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Improving neural machine translation models with monolingual data](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.
- Brian Thompson and Matt Post. 2020. [Paraphrase generation as zero-shot multilingual translation: Disentangling semantic similarity from lexical and syntactic diversity](#). In *Proceedings of the Fifth Conference on Machine Translation*, pages 561–570, Online. Association for Computational Linguistics.
- Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. [Bertscore: Evaluating text generation with bert](#). In *International Conference on Learning Representations*.

Evaluating Legal Accuracy of Neural Generators on the Generation of Criminal Court Dockets Description

Nicolas Garneau[†], Eve Gaumond[‡], Luc Lamontagne[†], and Pierre-Luc Déziel[‡]

Université Laval, Québec, Canada

Computer Science Department[†] and Faculty of Law[‡]

{nicolas.garneau, luc.lamontagne}@ift.ulaval.ca

eve.gaumond@observatoire-ia.ulaval.ca

pierre-luc.deziel@fd.ulaval.ca

Abstract

Docket files, also known as *plumitifs*, are legal text documents describing judicial cases. They are present in most jurisdictions and are meant to provide a window on legal systems. They contain information of a judicial case such as parties' identities, accusations' provisions, decisions, and pleas. However, this information is cryptic, using abbreviations, and making references to the criminal code. In this paper, we explore the use of neural text generators to improve the legal accuracy of the docket file verbalization regarding the accusations, decisions, and pleas sections. We introduce a legal accuracy evaluation scale used by jurists to manually assess the performance of three architectures with different levels of prior knowledge injection. We also study the correlation of our human evaluation methodology with automatic metrics.

1 Introduction

The *plumitif* [plymitif] is a legal registry providing short summaries of every judicial case heard by the courts in the province of Quebec, Canada. It is akin to what is known in English as court dockets, used in several other judicial systems. It provides information about the stakeholders involved in a case, the moment and the location where the various steps of the judicial process take place, and, in the case of the criminal *plumitif*, it also gives information about the offences, the pleas, and the verdicts. However, although this information is publicly available online, in reality, it is hardly accessible because of the format in which the *plumitif* is presented. It is written almost exclusively using abbreviations and makes numerous references to provisions in the Criminal code that are not defined anywhere (see Appendix A for an example). As a result, even experienced lawyers confess they sometimes have a hard time understanding the *plumitif* (Parada et al., 2020).

This lack of intelligibility is an issue (Tep et al., 2019; Parada et al., 2020; Beauchemin et al., 2020). Indeed, while the *plumitif* could serve many useful purposes, at the moment, it is not used to its full potential because of how hard it is to understand. For instance, the lack of intelligibility prevents self-represented litigants from using the *plumitif* to keep track of their cases. It also burdens the work of journalists using the *plumitif* to report on legal affairs. There are also instances of citizens who suffered prejudices because insurers misinterpreted their docket when they consulted it for background check purposes (Gaumond and Garneau, 2021). Therefore, tackling the issue of the understandability of Quebec's criminal *plumitif* is a worthy objective. It could promote access to justice, improve the transparency of the judicial system and prevent discrimination.

Beauchemin et al. developed a web application to tackle this issue. It works well to enhance the understandability of certain sections of the *plumitif* – the section about the parties involved in the case, for instance. However, the application, relying on a rule-based generator, lacks precision when it comes to generating a description of the charges. Indeed, it simply uses provisions' headings – as found in the Canadian Criminal code – to verbalize the charges. Hence, it would replace section 348 (1) of the Criminal code¹ with the following sentence: “Breaking and entering with intent, committing offence or breaking out.” This does not take into account the nuances of section 348 (1), which provides for four different offences;

1. “breaks and enters a place *with intent* to commit an indictable offence therein”
2. “breaks and enters a place and *commits* an indictable offence therein”

¹<https://laws-lois.justice.gc.ca/eng/acts/c-46/section-348.html>

3. “breaks out of a place *after*
 - (a) committing an indictable offence therein
 - (b) entering the place with intent to commit an indictable offence therein.

These four offences have different degrees of severity. For instance, a defendant breaking in somewhere with the intent of committing robbery could be remorseful and leave empty-handed the place he broke into. He would not be sanctioned as severely as another defendant who committed the robbery. Given the rule-based architecture [Beauchemin et al.](#) used, the only way for them to take more legal nuances into account would have been to “stitch” provision’s label with the corresponding paragraph and indent. Since a long stretch of text is known to be unintelligible ([Gaumont and Garneau, 2021](#)), this solution wasn’t suitable.

Instead, we propose to use neural architecture to generate descriptions of legal provisions that take legal nuances into account while being relatively concise. To that end, we trained neural text generators on *Plum2Text* ([Garneau et al., 2021c](#)), a Data-to-Text dataset, to solve this particular issue. However, neural architectures tend to hallucinate facts ([Dušek et al., 2018](#)) which raises a question regarding their usability to accomplish sensitive tasks such as ours. If these models were to hallucinate some information that does not appear in a docket – charges of which the defendant was not accused, for instance – they could not be used in a production setup.

In this paper, we propose a new legal accuracy evaluation scale used by jurists to manually assess the performance of the models we’ve trained. We analyze if they are accurate enough to be used in sensitive tasks such as verbalizing the content of the *plumitif*. We thus provide a comparative study of three neural architectures and reflect on their performance from a legal standpoint. We also evaluate them using automatic evaluation metrics and study their correlation with a human evaluation.

2 Training Neural Networks on *Plum2Text*

In this section, we introduce the three models we will evaluate. First, we introduce the *Plum2Text* dataset designed to train language generators, and then, we proceed to present the selected neural architectures as well as their training procedure.

2.1 *Plum2Text*

For our experiments, we have access to *Plum2Text*, introduced by [Garneau et al. \(2021b\)](#). It is a data-to-text dataset designed to train neural architectures to generate short descriptions of court dockets. It is derived from the pairings between criminal court judgments (a long textual document) and their associated docket file. A training instance is depicted in [Appendix B](#). *Plum2Text* contains 2,300 examples. The dataset is however heavily skewed towards common infractions such as “driving under the influence” (section 320.14 from the Canadian Criminal Code) and “assault and battery” (section 268). Our preliminary experiments showed that any neural text generator trained on *Plum2Text* *as-is* yielded models with poor generalization capabilities, often generating the most frequent offences. We thus undersampled *Plum2Text* so that every provision is represented by at most 5 examples. This undersampling yields a dataset of 1,602 examples that we split randomly into a train, valid, and test sets which contain 931, 247, and 424 examples respectively. To better assess the generalization capabilities of the generators, we identified 9 provisions in the test set that are neither in the train or valid sets. The provisions are listed in [Table 3](#), and we provide more details on the results of these specific examples in the evaluation [Section 3](#).

2.2 Neural Text Generators

Neural architectures have proven to be very effective at generating text in a wide variety of tasks. Long-Short Term memory networks using attention achieved impressive performance on automatic machine translation ([Bahdanau et al., 2015](#)). GPT, the Generative Pretraining architecture, based on the Transformer architecture ([Vaswani et al., 2017](#)), pushed automatic textual generation to a whole new level not only for machine translation but also for text summarization and data-to-text generation ([Radford et al., 2018, 2019](#); [Brown et al., 2020](#)). The performance of GPT is largely due to prior knowledge injection where fine-tuning on a downstream task requires less training data. Indeed, this model has been pre-trained on a large corpus before being trained on the target task. Prior knowledge injection is highly effective, especially when the prior is closer to the downstream task in terms of semantics and lexical field ([Howard and Ruder, 2018](#)). We thus consider three models, each with their respective degree of prior knowl-

edge injection. The first one is the model proposed by Bahdanau et al. (2015) trained from scratch on *Plum2Text (no prior)* using the same procedure as Wiseman et al. (2017). We then selected a French pre-trained model based on the Transformer network, BARThez (Kamal Eddine et al., 2021)² (*language prior*). The last model we consider is a fine-tuned version of BARThez on a legal corpus, *CriminelBART* (Garneau et al., 2021a) (*language and domain prior*).

In order to conduct our experiments, we used the fairseq library³ which provides implementations for the three models introduced in Section 2. For the three models, we used at most 1024 tokens (which resulted in batch sizes of 10 examples on average), the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.0005, a cosine learning rate scheduler, and a dropout of 0.1. The LSTM model converged after 15 epochs. BARThez converged after 2 epochs, and *CriminelBART* after only one epoch. We trained all models on a GeForce 2080 using a personal desktop. Each training takes less than an hour to run. The LSTM has around 3M parameters, while BARThez and *CriminelBART* have both 139M parameters. For the generation of legal descriptions, we used beam search, with a beam size of 6. We post-process the generations by detokenizing the sentences to increase the readability. At inference time, the LSTM model takes on average 0.1 seconds per generation, BART 1 second and *CriminelBART* 0.5 second. As a matter of reproducibility, we provide all the generations of the models here TODO. We provide generation examples in Appendix C.

3 Evaluation

As explained in section 1, the goal of this paper is to determine if neural architectures are accurate enough to be used in sensitive tasks such as verbalizing the content of the *plumitif*. Putting it another way, we want to see if some of the evaluated models could be used in a production setup. We also aim to characterize the strengths and weaknesses of neural generators in the field of law. We first introduce our methodology, discuss our expectations and finally analyze the results.

²BARThez is the French version of BART (Lewis et al., 2020), which showed interesting performance across several datasets for the task of data-to-text generation (Gehrmann et al., 2021).

³<https://github.com/pytorch/fairseq>

3.1 Methodology

We first introduce new human evaluation guidelines motivated by the underlying task of measuring the legal accuracy of the models. We then analyze the performance of the models using several automatic evaluation metrics. Finally, we analyze the correlation between automatic and human evaluation in order to ground one or several metrics in the context of automatic model selection.

3.1.1 Human Evaluation

Generating descriptions of criminal court dockets – which we trained our neural-text-generators for – is a rather sensitive task. Inaccurate generations run contrary to the very objective we pursue and could have real consequences. For example, imagine the potential harms resulting from a docket description that says that a defendant is guilty of a charge, while he was actually acquitted; that he was accused of possessing child pornography while he was actually accused of possession of cannabis; or that he pleaded guilty while it was not the case. This is why we deemed it essential to assess the quality of the generations not only from a technical standpoint but also from a legal standpoint. Following the arguments of van der Lee et al. (2019), we answer several questions regarding the experimental setup and the choices we made.

Selected models. We selected all three models trained on *Plum2Text*, allowing us to evaluate the improvement of prior knowledge injection in the field of legal text generation.

Number of outputs. From the selected test set containing 232 instances, we carefully selected instances yielding a diverse sample for the annotators to evaluate. This resulted in 89 instances, 64 with one table value and 17 with two table values. We manually created 8 instances containing three table values (*i.e.*, provision, decision, and pleading) since no instance contain the three different types of values in the original test set. Each instance is associated with three output generations, yielding a total of 267 outputs to evaluate.

Input selection. Following the recommendations of van Miltenburg et al. (2021), we select specific kinds of inputs and analyze their corresponding outputs. Hence, we begin by presenting to the annotator simple inputs containing only one table value. We then gradually increase the complexity of the inputs going up to three table values,

which represent a whole *plumitif*'s line (charge, pleading, verdict, except for the sentence). This procedure allows the annotators to become familiar with the annotation interface, the dataset, and the task. We can also analyze the performance of the models given the inputs' increasing complexity.

Presentation and interface. We used the Prodigy annotation tool (Montani and Honnibal, 2018) and customized it for our need to present the *plumitif*'s input data and the three models' outputs. For each instance, the outputs are randomly ordered. The annotators are asked to score each of the three models' generation independently. This way of characterizing generations' relevance simultaneously has proven to be highly efficient in a model selection setup (Novikova et al., 2018). The evaluation interface is illustrated in Figure 1.

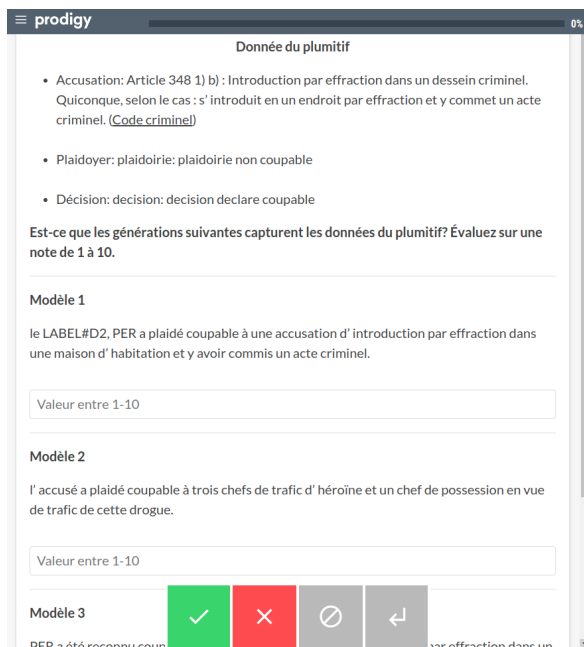


Figure 1: The Prodigy annotation interface used by the annotators to semantically evaluate the generation of the three neural architectures.

Annotators. We selected three annotators with legal knowledge to evaluate the generations. The first annotator is a coauthor of this paper. She holds a bachelor's degree in law. Since she mainly works on the legal aspects of the project, she had not seen any of the generations nor any of the models before conducting the evaluations. She advised the principal investigator in the drafting of the evaluation guidelines and went through the evaluation process before the two other annotators to ensure that the guidelines were sufficiently clear for people trained

in law. Her results should be read with all of that in mind. The other two annotators are second-year law students at the Faculty of Law. They were introduced to the context, the task, and the annotation interface in a meeting with the principal investigator and the first annotator. Another meeting was also held after a pilot evaluation. During this meeting, annotators 2 and 3 – who by then had evaluated 5 instances (i.e. 15 generations) – received feedback and advice on what phenomena they should be careful for. Annotators are paid at an hourly rate of 17 CAD/hour. Annotators were asked to spend at most 5 minutes per instance. It took a total of 8 hours for each annotator to complete the evaluation, including the training, the pilot and reading the evaluation guidelines.

After the annotators completed the evaluation, we gathered their comments on the difficulty of the task and if they encountered ambiguous cases. It turned out that the provisions' texts can be ambiguous since they may contain some disjunction in regards to the committed offence. Take for example provision 320.14 (1) a), "Operation while impaired", which states that "Everyone commits an offence who operates a conveyance while the person's ability to operate it is impaired to any degree by *alcohol or a drug or by a combination of alcohol and a drug*;"'. For this provision, models always generated a description only regarding the "degree of alcohol", omitting the drug aspect of the offence. However, one would need to look at the judgment file (if any) to validate if the defendant operated the conveyance impaired by alcohol or a drug or a combination of both. In such cases, annotators were unsure if the generation contained hallucinated/omitted facts, since the generation was not totally supported by the docket file's data. We can thus conclude that given a high agreement score and several ambiguous cases suggest that our instruction were clear for the annotators and the legal accuracy scale was easy to use. We provide in-depth details of the evaluation setup in our Human Evaluation Datasheet (Shimorina and Belz, 2021) in Appendix E.

Legal Accuracy Scale. Given that our aim with this paper is to determine whether or not neural-text generators are sufficiently accurate to be used in a production setup, we needed a definition of the notion of legal accuracy for our particular context as well as an assessment tool to measure it. We thus define "legal accuracy" as a metric that measures

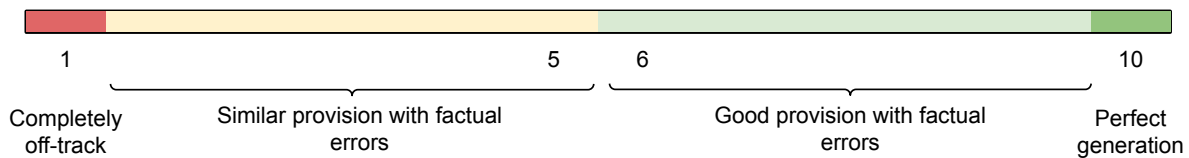


Figure 2: The legal accuracy scale used by the human annotators. The annotators first decide where, between the four regions (completely off-track, similar provision, good provision, and perfection generation) the actual generation sits. Then they remove points for every hallucination and/or omissions encountered.

the congruence between the docket’s description that our models generate and the input data (Criminal code provision, plea, and verdict) the model aims to describe. To this end, we designed a Likert scale ranging from one to ten (Likert, 1932). A generation that scores a ten is highly accurate. However, a generation receiving a score of 1 misses the mark as it does not match the input data at all. To determine the legal accuracy score that a generation should receive, the annotator has to follow a two-step process. First, they decide if the docket description is 1) accurate, 2) thematically relevant, or 3) off-track. The legal accuracy scale is split into three regions;

1. **6 - 10 – Accurate.** If the generation refers to the good provision, it is considered accurate and will score be between 6 and 10.
2. **2 - 5 – Thematically Relevant.** If the generation is “on theme” with the input data, the score will be between 2 and 5. A thematically relevant description is related to the right provision, but not perfectly on point (possession of drugs vs possession of weapons; sexual exploitation vs child pornography; breaking in with the intent of committing a crime vs breaking in and committing a crime).
3. **1 – Off-Track.** If the generation is about “Mischief” while the input was about “Drug trafficking”, we ask the annotator to assign a score of 1 since it is completely off-track.

Once the annotators have chosen the bracket where the generation belongs (1; 2-5; 6-10)⁴, they can start moving on to the second step: looking for factual errors. We identify three types;

⁴ At first, we split the scale into four regions: 1; 2-5; 6-9; 10. However, the annotators tend to naturally split it into three regions since they can not directly attribute 10 points to a given generation whereas they can directly attribute 1 point to an irrelevant generation. They first need to see if the generation is accurate, on theme, or irrelevant, before proceeding to the second step.

1. **Hallucinations:** facts that the model generates even though it does not appear in the input data. There are various kinds of hallucinations: the model generates a charge, verdict, or plea that does not appear in the *plumitif*, provides some factual details about the perpetration of the infraction that should not appear in the *plumitif* (e.g. the defendant did traffic *heroin* unlawfully). One point should be removed per hallucination.
2. **Omissions:** occurs when facts are in the input data but end up not being generated by the model. One element that is quite often omitted is the provision number. The absence of the plea or the verdict is also considered an omission when it was available in the input data. One point should be removed per omission.
3. **Confusions:** factual mistakes characterized by the mismatch of the input data and the content of the generation. For example, the input data says that the defendant pleaded guilty while he appears to have pleaded not guilty in the generation, or the court orders a stay of procedures in the *plumitif*, and the defendant is found guilty in the generation. In these cases, two points should be removed: one for hallucinating a fact, and one for omitting a fact.

No matter how many factual errors there are, they can’t make a generation downgrade to the lower bracket. So, a thematically relevant generation can’t have less than 2 points, and a generation that gets the provision right can’t have less than 6 points. Finally, a provision from the 6-10 points bracket which is exempt from factual errors, gets 10 points, which is the highest possible mark on our legal accuracy scale. To summarize this process, Figure 2 provides a conceptual illustration of the Likert scale.

3.1.2 Automatic Evaluation

For the automatic evaluation of the neural models, we use the same set of commonly used metrics as in the GEM benchmark suite (Gehrmann et al., 2021). We can differentiate the metrics according to two features: those using surface tokens or vector representations, and those using the reference and/or the table values. BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), METEOR (Banerjee and Lavie, 2005) use surface tokens and a reference. BertScore (Zhang* et al., 2020) (using the underlying multilingual version of BERT (Devlin et al., 2019)⁵) uses vector representation and a reference. We also consider two metrics using vector representation and the table values recently introduced by Dušek and Kasner (2020) and Garneau and Lamontagne (2021), which we dubbed respectively NLI and RANK in this paper.

NLI uses natural language inference to check if a given hypothesis entails or contradicts table values. According to the methodology described by Dušek and Kasner (2020), we created three templates needed for the computation of the NLI metric. These three templates are each associated to a specific type of value, which are the accusation, the plea, and the verdict that take as input a subject and an object⁶;

- <subject> is accused of <object> .
- <subject> pleaded <object> .
- <subject> is declared <object> .

Given the table values, we fill in these templates and perform natural language inference w.r.t. the hypothesis under test. We used the pre-trained CamemBERT (Martin et al., 2020) base NLI model in our experiments.

RANK uses a ranking model coupled with the mean average precision to assess the ability of a given hypothesis to retrieve its corresponding table values. According to the methodology described by Garneau and Lamontagne (2021), we trained the multilingual version of BERT using the *plum2text* dataset on the semantic textual similarity task. This yielded a model able to rank table values w.r.t. the hypothesis under test, as required by RANK.

A reference-less metric can be interesting in cases where we do not have access to manually

⁵We did not use BLEURT (Sellam et al., 2020) since it has been trained on an English corpus.

⁶The subject is always the same, being “*The defendant*”

curated pairs of table–reference or in a production setting where references are simply nonexistent. As Zhang* et al. (2020) suggest, metrics using embeddings instead of surface tokens showed better correlation with human evaluation in several settings, a phenomenon we wish to confirm in our setup.

3.1.3 Grounding Metrics

Finally, we wish to ground automatic evaluation metrics w.r.t the legal accuracy scores to speed up the model selection process, which would be highly desirable in a concrete application setup (Belz and Reiter, 2006; van der Lee et al., 2019). To this end, we compute the Spearman correlation of the human evaluation scores with every automatic metric introduced in the previous section.

3.2 Expectations

According to the goal and the human and automatic evaluation methodologies previously introduced, we have the following expectations regarding the experiments;

1. We expect that models containing more prior knowledge on the downstream task will perform better and may have better generalization capabilities, as exposed by (Peters et al., 2018; Devlin et al., 2019; Raffel et al., 2020). This supports our approach of using the models mentioned previously with three different levels of prior knowledge.
2. We do not expect high correlation scores between human evaluation and metrics based on word overlap (BLEU, ROUGE, METEOR) (Belz and Reiter, 2006; Novikova et al., 2017). However, we expect better correlation scores with metrics that use vector representations (BertScore) and use the input table for their computation (NLI, RANK) (Zhang* et al., 2020).
3. We expect that the increasing complexity of the input (i.e. adding the verdict and the plea as input) should not impact the models’ performance dramatically since the range of values of this type of data is limited (e.g. up to 10 different verdicts and two different pleas).

3.3 Results

In this section, we first analyze how automatic metrics correlate with human judgment. We then study

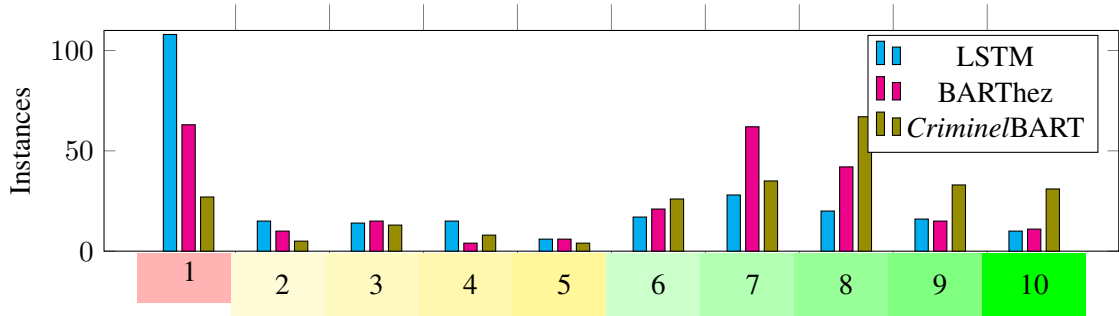


Figure 3: Results of the human evaluation according to the legal accuracy scale. We present the results of the vanilla LSTM (no prior), BARThez (language prior), and *CriminelBART* (language and domain prior).

	LSTM	BARThez	<i>CriminelBART</i>
Ann. 1	4.4±2.8	5.2±2.9	6.3±2.6
Ann. 2	3.7±3.2	5.2±3.0	6.8±2.8
Ann. 3	3.6±3.3	5.4±3.2	7.0±2.8
Avg.	3.9±2.9	5.3±2.9	6.7±2.6
ρ	0.76	0.85	0.84

Table 1: Average score and standard deviation per annotator and the overall score for each model. We also provide the annotator agreement ρ per model. The overall agreement is 0.84.

the benefit of language and domain prior knowledge injection, both on seen and unseen distributions of the data. We also diagnose the learning dynamics of the neural architecture w.r.t the increasing complexity of the input.

3.3.1 Prior knowledge

Results of the human evaluation on the 267 outputs are displayed in Figure 3 using the legal accuracy scale. We can see that the LSTM model has difficulty finding itself on the right side of the scale, having more than 100 irrelevant generations and achieving an overall score of 3.9. BARThez, containing a substantial language prior, does perform much better than the vanilla LSTM, achieving an average score of 5.3 mostly due to its 60 irrelevant generations. Its generations are mostly spread on the far left, and middle right of the scale. *CriminelBART* achieves the best performance with an overall score of 6.7, having most of its generation containing the “good provision”. From these results, and w.r.t. the legal accuracy scale, this tells us that on average, *CriminelBART* will be on theme with possibly 2-3 hallucinations/omissions. This observation validates our first expectation regarding the contribution of prior knowledge injection.

We also provide the breakdown of the scores by annotator in Table 1. Annotator 1 provided scores on a narrow scale, ranging from 4.4 to 6.3 on average, whereas Annotator 2 and 3 used a wider scale with scores ranging from 3.6 to 7.0. Since we have multiple annotators and an ordinal scale, we used Krippendorff’s alpha coefficient (Krippendorff, 2004) to measure inter-annotator agreement. We obtained a correlation coefficient ρ of 0.84 across all models. This high correlation coefficient suggests that either the evaluation task was easy and/or the evaluation guidelines were clear and easily understood by the annotators. Looking at the agreement model-wise, we obtained a ρ coefficient of 0.85 and 0.84 for the BARThez and *CriminelBART* evaluations, respectively. For the LSTM model, we obtained a ρ coefficient of 0.74. It seems like the annotators tend to disagree when the generations are worse, probably misclassifying a generations as being “on theme” (2-5) or “irrelevant” (1).

In Table 2, we present the results of the experiments using automatic evaluation metrics. One of our expectations was that the more prior knowledge a model has, the better it will perform. While *CriminelBART* is the best model across all metrics, it is interesting to see however that, according to the metrics using references, BARThez performs worse than the vanilla LSTM. On the other hand, by looking at the metrics using the table values, BARThez seems to be substantially better than the LSTM model. From these results, it is not clear if the language prior was truly beneficial in our setup. However, the domain prior improves substantially the performance of the generations.

Finally, we analyze the generalization capabilities of the models on *unseen provisions* i.e., provisions that were included neither in the training

	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR	BERTScore	NLI	RANK
LSTM	0.38	0.28	0.23	0.20	0.33	0.20	0.75	0.28	0.21
BARThez	0.32	0.24	0.19	0.16	0.34	0.21	0.74	0.34	0.38
<i>Criminel</i> BART	0.51	0.42	0.36	0.32	0.44	0.28	0.78	0.34	0.43

Table 2: Automatic evaluation results of the three models using token-based metrics (BLEU, ROUGE, and METEOR) and embedding-based metrics (BERTScore, NLI, and RANK).

Provision	LSTM	BARThez	<i>Criminel</i> BART
445.1 (1) a)	1.0	1.0	1.0
150	2.3	5.0	4.6
83.181	1.0	1.0	1.0
241	1.0	2.7	2.0
467.12	1.0	1.0	8.7
810.2	1.0	1.0	1.0
172	1.0	1.0	1.33
320.14	1.0	6.3	7.3

Table 3: Analysis of the generalization capabilities of the models on unseen provisions. We provide details on the provisions in Appendix D.

nor in the validation sets. We identified 8 unseen provisions, listed in Table 3. The results show that all the models struggle to fully generalize to unseen provisions. We can see that the LSTM can not generalize to unseen provision, which is expected. An interesting fact is that even if BARThez does not have any domain prior, it generalizes as well as *Criminel*BART except for one provision, 467.12, which corresponds to “Commission of offence for criminal organization”. While BARThez and *Criminel*BART achieve a decent performance on provision 320.14 (Operation while impaired), it is more of a training set artifact since provision 253 that has been repealed in 2018 also corresponding to “Operation while impaired” is present in the training set. In a similar vein, the repealed provision 150 corresponding to “having illegally in his possession for sale magazines that are obscene” is similar to several many other charges of a sexual nature in the training set (e.g. 163, “Obscene materials”) explaining why every model are “on-theme” for this provision.

3.3.2 Correlations Between Human and Automatic Evaluations

In every case, results show a positive correlation between human evaluation and automatic metrics. Word overlap metrics (BLEU- x , ROUGE-L,

and METEOR) tend to show decreasing correlation scores as the model produces better generations; going from 0.4 with the LSTM to 0.2 with *Criminel*BART. BERTScore, an embedding-based metric, presents a high correlation score with the LSTM model. However, regarding BARThez and *Criminel*BART, correlation scores drop as low as 0.12. NLI provides consistent correlation scores of 0.35 on average, regardless of the model. RANK offers the highest correlation scores w.r.t. the models, reaching 0.81 with the LSTM model, 0.62 with BARThez, and 0.40 with *Criminel*BART. We suppose that these high correlation scores are tied to the nature of the last two metrics; they are using the input values as a way to assess the relevance of the generation, thus measuring its factual accuracy. On the other hand, overlap-based metrics and BERTScore only use the target reference which may not capture the factual accuracy one may be looking for. In light of these results, we deem it possible to use one or several metrics grounded with the proposed human evaluation to select the best-performing model for futur works.

3.3.3 Increasing Complexity of the Input

To better understand the learning dynamics of the neural architectures, we analyze their performance w.r.t. to the increasing complexity of the input i.e., going from one to three table values. More precisely, we want to study how models are able to combine semi-structured information that has been “linearized” as in Wiseman et al. (2017). Looking at Table 5, we can see that the performance of the LSTM model rapidly decreases as we add values in the input, going from 4.8 with one value, to 2.0 with two, and 1.0 with three. Unfortunately, the model is not able to generate relevant descriptions as the complexity increases. We can also see a slight decrease in performance with the BARThez model going from 5.7 to 4.6 and 3.9 for one, two, and three input values respectively. *Criminel*BART, on the other hand, did maintain relevant generations with two input values with an average score of

	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR	BERTScore	NLI	RANK
LSTM	0.39	0.41	0.37	0.36	0.45	0.39	0.39	0.32	0.81
BARThez	0.25	0.31	0.31	0.28	0.38	0.40	0.12	0.35	0.62
<i>Criminel</i> BART	0.20	0.21	0.20	0.19	0.26	0.28	0.25	0.30	0.40

Table 4: Spearman correlation scores of automatic metrics with human evaluation. All scores have a p -value < 0.05 except for the pairs BARThez–BERTScore and *Criminel*BART–BLEU- x , which exhibit the lowest correlations. We highlighted in bold “row-wise” highest correlations, showing that RANK has capabilities to select the best model.

	LSTM	BARThez	<i>Criminel</i> BART
1 Value	4.8±2.9	5.7±2.8	6.8±2.7
2 Values	2.0±1.0	4.6±2.9	7.3±1.9
3 Values	1.0±0.0	3.9±2.8	4.5±1.7

Table 5: Analysis of the increasing complexity of the input by models, going from one to three table values.

7.3. However, its performance decreases with three table values, dropping at 4.5 on average. This analysis suggests that generating the complete line of a docket file (*i.e.*, accusation, decision, and pleading) is not properly handled by the neural architectures and that more training data would be beneficial. This observation invalidates our expectation that adding the verdict and the plea does not impact the models’ performance.

4 Conclusion

In this paper, we evaluated the performance of three neural architectures, both automatically and manually, on the Data2Text task of docket files description generation. We proposed a new 10-point Likert scale to assess the legal accuracy of these architectures. We studied the correlation of automatic metrics with our human evaluation methodology and found out that the RANK metric can be used for automatic model selection. We release the generations of all three models as well as their associated automatic and human (anonymous) evaluation scores for a matter of reproducibility and for the research community’s benefit. Unsurprisingly, *Criminel*BART is the best performing model due to its prior knowledge of the legal field. On average, it generates descriptions containing the good provision and better handles the increasing complexity of the input. However, its hallucination and omission rates suggest the need for improvements in this regard to obtain acceptable legal accuracy. Future works will look at better ways to condition this model to improve its legal accu-

racy using hard constraints (Meister et al., 2020) and post-edition (Mallinson et al., 2020). However, we believe that these models will require a human validation to be used in production, due to their inherent probabilistic nature and the sensitive legal field. We further discuss this matter, as well as the ethical considerations of having such a model in production in the following Section 5; **Broader Impacts – Law and Ethics.**

Acknowledgements

We thank the reviewers for their insightful comments as well as David Beauchemin, Ulysse Côté-Allard, Mathieu Godbout, Jean-Samuel Leboeuf, and Frédéric Paradis from the *Groupe de recherche en apprentissage automatique de Laval* who peer-reviewed our manuscript. This research was enabled in part by the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Social Sciences and Humanities Research Council (SSHR). Also, this research was made possible with the help and contribution of our partner, *La Société Québécoise d’Information Juridique*. Lastly, we wish to thank the annotators who evaluated the generations, Andréa Lampron and Noémie Lamothe.

5 Broader Impacts – Law and Ethics

As discussed in the introductory part, Quebec’s *plumitif* is hard to understand. This well-documented issue (Parada et al., 2020; Tep et al., 2019; Prom Tep et al., 2020; Beauchemin et al., 2020) hinders access to justice, causes prejudices to people subject to background checks and contributes to a certain opacity of the judicial system (Gaumond and Garneau, 2021). Beauchemin et al. developed a web application to address this issue, but the performance of their rule-based text-generator is not satisfactory w.r.t the description of the charges. We thought that an alternative architecture, based on neural networks, could improve the charges’ description. However, we were uncertain about the legal accuracy of neural-text generators knowing their propensity to hallucinate facts (Dušek et al., 2018). Therefore, we designed an evaluation method to assess the legal accuracy of three neural models generating descriptions of criminal charges. This process leads to the conclusion that *CriminelBART* is – with an average score of 6.7/10 – the best model to generate descriptions of criminal charges appearing in Quebec’s *plumitif*. In the next sections, we reflect on what is required, in terms of legal accuracy.

5.1 What Is Considered Accurate Enough?

AI technologies used in the legal system ought to reach a high level of accuracy. This is obvious when we think about predictive tools informing judges’ decisions (Surden, 2020) such as COMPAS, the infamous recidivism prediction algorithm (Dressel and Farid, 2018). But it should be equally clear that accuracy is crucial for AI systems used to disseminate judicial information. The intended purpose of an AI system determines the level of accuracy it should meet. *CriminelBART* aims at reducing the number of errors people make when they access the *plumitif*. It’s a purpose that commands a high degree of accuracy. Indeed, if its generations are inaccurate, *CriminelBART* is both useless and dangerous. Useless because it goes against the very purposes it tries to achieve; and dangerous because providing inaccurate information about people’s criminal history could lead to harm such as discrimination.

5.2 Is *CriminelBART* accurate enough?

We voluntarily chose not to pinpoint where the legal accuracy threshold falls; we don’t want to say

that a score of 9.5 on our scale means that a model is ready for production. Determining if a model is ready to move to production is contextual. A specific risk assessment should be done to make such a determination. In this case, the conclusion is that *CriminelBART* isn’t accurate enough. With an average of two or three factual mistakes per generation – and even more inaccuracies when it comes to unseen provisions – *CriminelBART* is not ready to be used in a production setup. The example below provides an illustration:

- On REDACTED DATE, at REDACTED PLACE, the defendant broke and entered a dwelling-house with the intention to commit an offence therein, thus committing the indictable offence provided at section 348(1)b)d) of the Criminal Code.

There are four problems with this generation. First, there is one offence – sexual assault – that doesn’t appear in the generation even though it was inputted into the model. Second, the generation says that the break-in happened in a dwelling-house while no such information was input into the model. This hallucination could be consequential since break-ins in dwelling-houses are considered more serious, and receive longer sentences. Third, the date and location of the offence are also hallucinated. Finally, the provision number should have been 348(1)b) instead of 348(1)b)d).

5.3 How to Increase Legal Accuracy?

Given the high degree of accuracy required for our purposes, it is not clear that neural text-generators will ever be accurate enough to be used without human oversight. Combining computers and humans’ strength to increase *CriminelBART*’s accuracy might be the way forward. Since writing descriptions of the Criminal code’s provisions is a tedious task unlikely to be undertaken by humans, *CriminelBART* could generate drafts that court clerks would post-edit for accuracy. However, clerks are already tied-up. To ensure the adoption of the technology, this new post-edition task shouldn’t feel burdensome to them. Players in the field make the success of legal innovations. It’s important to make sure that their opinion is heard and considered and that they see the innovation as presenting some advantages for them (Benyekhlef et al., 2016).

References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.
- Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. [Abstract Meaning Representation for sembanking](#). In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.
- Satanjeev Banerjee and Alon Lavie. 2005. [METEOR: An automatic metric for MT evaluation with improved correlation with human judgments](#). In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.
- David Beauchemin, Nicolas Garneau, Eve Gaumond, Pierre-Luc Déziel, Richard Houry, and Luc Lamontagne. 2020. [Generating intelligible plunitifs descriptions: Use case application with ethical considerations](#). In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 15–21, Dublin, Ireland. Association for Computational Linguistics.
- Anja Belz and Ehud Reiter. 2006. [Comparing automatic and human evaluation of NLG systems](#). In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy. Association for Computational Linguistics.
- Karim Benyekhlef, Jane Bailey, Jacquelyn Burkell, and Fabien Gelin, editors. 2016. *eAccess to Justice*. Law, Technology and Media. University of Ottawa Press, Ottawa, ON, Canada.
- Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.
- Dallas Card, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky. 2020. [With little power comes great responsibility](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9263–9274, Online. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Julia Dressel and Hany Farid. 2018. [The accuracy, fairness, and limits of predicting recidivism](#). *Science Advances*, 4(1).
- Ondřej Dušek and Zdeněk Kasner. 2020. [Evaluating semantic accuracy of data-to-text generation with natural language inference](#). In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 131–137, Dublin, Ireland. Association for Computational Linguistics.
- Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2018. [Findings of the E2E NLG challenge](#). In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 322–328, Tilburg University, The Netherlands. Association for Computational Linguistics.
- Nicolas Garneau, Eve Gaumond, Luc Lamontagne, and Pierre-Luc Déziel. 2021a. [Criminelbart: A french canadian legal language model specialized in criminal law](#). In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law, ICAIL '21*, page 256–257, New York, NY, USA. Association for Computing Machinery.
- Nicolas Garneau, Eve Gaumond, Luc Lamontagne, and Pierre-Luc Déziel. 2021b. [Plum2text: A french plunitifs-descriptions data-to-text dataset for natural language generation](#). In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law, ICAIL '21*, page 200–204, New York, NY, USA. Association for Computing Machinery.
- Nicolas Garneau, Eve Gaumond, Luc Lamontagne, and Pierre-Luc Déziel. 2021c. [Plum2text: A french plunitifs-descriptions data-to-text dataset for natural language generation](#). In *Proceedings of the 18th International Conference on Artificial Intelligence and Law*, Sao Paulo, Brazil. International Association for Artificial Intelligence and Law.
- Nicolas Garneau and Luc Lamontagne. 2021. [Trainable ranking models to evaluate the semantic accuracy of data-to-text neural generator](#). In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 51–61, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Eve Gaumond and Nicolas Garneau. 2021. [5q ia clartÉ plumcr qc : Cinq questions permettant d’appréhender l’usage d’intelligence artificielle pour accroître la clarté du plunitif criminel québécois](#). In *Lex Electronica*, editor, *La justice dans tous ses états*, pages 216–248. Montréal.
- Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu,

- Dipanjan Das, Kaustubh Dhole, Wanyu Du, Esin Durmus, Ondřej Dušek, Chris Chinenye Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Mihir Kale, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Andre Niyongabo Rubungo, Salomey Osei, Ankur Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. 2021. [The GEM benchmark: Natural language generation, its evaluation and metrics](#). In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 96–120, Online. Association for Computational Linguistics.
- Jeremy Howard and Sebastian Ruder. 2018. [Universal language model fine-tuning for text classification](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.
- David M. Howcroft, Anya Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A. Hasan, Saad Mahamood, Simon Mille, Emiel van Miltenburg, Sashank Santhanam, and Verena Rieser. 2020. [Twenty years of confusion in human evaluation: NLG needs evaluation sheets and standardised definitions](#). In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 169–182, Dublin, Ireland. Association for Computational Linguistics.
- Moussa Kamal Eddine, Antoine Tixier, and Michalis Vazirgiannis. 2021. [BARThez: a skilled pretrained French sequence-to-sequence model](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9369–9390, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Klaus Krippendorff. 2004. *Content Analysis: An Introduction to Its Methodology (second edition)*. Sage Publications.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Rensis Likert. 1932. *A technique for the measurement of attitudes*. Archives of psychology ; no. 140. [s.n.], New York.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Jonathan Mallinson, Aliaksei Severyn, Eric Malmi, and Guillermo Garrido. 2020. [FELIX: Flexible text editing through tagging and insertion](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1244–1255, Online. Association for Computational Linguistics.
- Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. [CamemBERT: a tasty French language model](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.
- Clara Meister, Ryan Cotterell, and Tim Vieira. 2020. [If beam search is the answer, what was the question?](#) In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2173–2185, Online. Association for Computational Linguistics.
- Ines Montani and Matthew Honnibal. 2018. Prodigy: A new annotation tool for radically efficient machine teaching.
- Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. [Why we need new evaluation metrics for NLG](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252, Copenhagen, Denmark. Association for Computational Linguistics.
- Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2018. [RankME: Reliable human ratings for natural language generation](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 72–78, New Orleans, Louisiana. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: A method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02*, page 311–318, USA. Association for Computational Linguistics.

- Alexandra Parada, Sandrine Prom Tep, Florence Millerand, Pierre Noreau, and Anne-Marie Santorineos. 2020. Digital Court Records : a Diversity of Uses. *IJR*, 9.
- Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [Deep contextualized word representations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.
- Sandrine Prom Tep, Florence Millerand, Alexandra Bahary-Dionne, Sarah Bardaxoglou, and Noreau Pierre. 2020. Le “plumitif accessible” : les enjeux liés à l’accès aux registres informatisés en ligne. In Yvon Blais, editor, *22 chantiers pour l’accès au droit et à la justice*, pages 43–66. Montréal.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. [BLEURT: Learning robust metrics for text generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.
- Anastasia Shimorina and Anya Belz. 2021. The human evaluation datasheet 1.0: A template for recording details of human evaluation experiments in nlp. *ArXiv*, abs/2103.09710.
- Harry Surden. 2020. [Ethics of AI in law](#).
- Sandrine Prom Tep, Florence Millerand, Alexandra Parada, Alexandra Bahary, Pierre Noreau, and Anne-Marie Santorineos. 2019. Legal Information in Digital Form: the Challenge of Accessing Computerized Court Records. *IJR*, 8.
- Chris van der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Kraemer. 2019. [Best practices for the human evaluation of automatically generated text](#). In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 355–368, Tokyo, Japan. Association for Computational Linguistics.
- Emiel van Miltenburg, Miruna Clinciu, Ondřej Dušek, Dimitra Gkatzia, Stephanie Inglis, Leo Leppänen, Saad Mahamood, Emma Manning, Stephanie Schoch, Craig Thomson, and Luou Wen. 2021. [Underreporting of errors in NLG output, and what to do about it](#). In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 140–153, Aberdeen, Scotland, UK. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.
- Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. [Challenges in data-to-document generation](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.
- Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. [Bertscore: Evaluating text generation with bert](#). In *International Conference on Learning Representations*.

A Example Docket File

An example docket file is depicted in Figure 4. The accusation section, starting at the middle of the document, contains provisions' number, paragraph and indent (*465(01)c) in the figure) as well as the associated decision and plea (PLAIDOYER NON COUPABLE and DECISION DECLARE COUPABLE).

```

                                SEQ.ACC. 001/001
ACC. 3560
    3560 QUEBEC, QUEBEC
    NAIS
    AVO.
                                DATE INFRACTION
                                DATE OUVERTURE
PLA. 1130,
    1130, QUEBEC (QUEBEC)
    AVO.
ORG. SERVICE DE POLICE DE LA VILLE
    NO.

2 CHEFS D'ACCUSATION

CODE CRIMINEL          FED
01 01 *465(01)C)
    12:50 PLAIDOYER NON COUPABLE
    12:05 DECISION DECLARE COUPABLE
    09:27 PEINE
TEMPS PASSE SOUS GARDE: 220 JOURS
PERIODE INFLIGEE SANS PROVISoire: 6 MOIS
PEINE INFLIGEE DE 6 MOIS
PROBATION DE 36 MOIS SURV. PROBATION SUIVI 12 MOIS
240H T.C. DELAI:15 MOIS
```

Figure 4: A *Plumitif* document presenting the defendant and plaintiff personal information along with charges and associated pleas, decisions and penalty. Some regions have been blurred for privacy concerns.

B Plum2Text Training Instance

A training instance from the *Plum2Text* training dataset pictured in Figure 5.

Table values

Accusation: Provision 320.14 (1) a)
Every person commits an offence who : (a) operates a conveyance while his or her ability to drive is impaired to any degree by the effect of alcohol or a drug or by the combined effect of alcohol and a drug;
Plea
Pleaded not guilty
Decision
Declared guilty

Reference

PER pleaded not guilty on a count of impaired driving and was declared guilty.
--

Figure 5: An instance from *Plum2Text*' training set containing three table values with its associated reference. Text has been translated from French to English.

Input Data	<u>accusation</u> : 348 1) a) Introduction par effraction dans un dessein criminel: quiconque, selon le cas: s'introduit en un endroit par effraction avec l'intention d'y commettre un acte criminel; <u>plaidoyer</u> : coupable.
LSTM	l'accusé a plaidé coupable à trois chefs de trafic d'héroïne et un chef de possession en vue de trafic de cette drogue.
BARThez	PER a plaidé coupable à des accusations de s'être introduit par effraction dans une maison d'habitation , de s'être livré à des voies de fait, de s'être évadé, de s'être livré à des voies de fait, de s'être livré à des voies de fait et de s'être livré à des actes criminels.
CriminelBART	PER plaide coupable à une accusation d'introduction par effraction dans une maison d'habitation avec l'intention d' y commettre un acte criminel.

Table 6: Example generations from the three models on the input data of provision 348 1) a) and a guilty plea. We can see that the LSTM is completely off-track (drug trafficking) while BARThez hallucinates several facts (assault and escaped from jail). *CriminelBART* contains the good provision (breaking and entering with the intent to commit a crime), but hallucinates “in a dwelling house”.

C Generation Examples

Table 6 presents example generations from the three models given the input “provision 348 and a guilty plea”.

D Unseen Provisions

- **445.1 (1) a):** *“Causing unnecessary suffering. Every one commits an offence who wilfully causes or, being the owner, wilfully permits to be caused unnecessary pain, suffering or injury to an animal or a bird;”*
- **150:** *“Illegally had in his possession for sale magazines that are obscene.”*
- **83.181:** *“Leaving Canada to participate in activity of terrorist group. Every person who leaves or attempts to leave Canada, or goes or attempts to go on board a conveyance with the intent to leave Canada, for the purpose of committing an act or omission outside Canada that, if committed in Canada, would be an offence under subsection 83.18(1) is guilty of an indictable offence and liable to imprisonment for a term of not more than 10 years.”*
- **241 (1) a):** *“Counselling or aiding suicide. Everyone is guilty of an indictable offence and liable to imprisonment for a term of not more than 14 years who, whether suicide ensues or not, counsels a person to die by suicide or abets a person in dying by suicide;”*
- **811 a):** *“Breach of recognizance. person bound by a recognizance under any of sections 83.3 and 810 to 810.2 who commits a breach of the recognizance is guilty of an indictable offence and is liable to imprisonment for a term of not more than four years;”*
- **467.12 (1):** *“Commission of offence for criminal organization. Every person who commits an indictable offence under this or any other Act of Parliament for the benefit of, at the direction of, or in association with, a criminal organization is guilty of an indictable offence and liable to imprisonment for a term not exceeding fourteen years.”*
- **810.2:** *“Where fear of serious personal injury offence. Any person who fears on reasonable grounds that another person will commit a serious personal injury offence, as that expression is defined in section 752, may, with the consent of the Attorney General, lay an information before a provincial court judge, whether or not the person or persons in respect of whom it is feared that the offence will be committed are named.”*
- **172 (1) a):** *“Corrupting children. Every person who, in the home of a child, participates in adultery or sexual immorality or indulges in habitual drunkenness or any other form of vice, and by doing so endangers the morals of the child or renders the home an unfit place for the child to be in, is guilty of an indictable offence and liable to imprisonment for a term of not more than two years;”*
- **320.14 (1) a):** *“Operation while impaired. Everyone commits an offence who operates a conveyance while the person’s ability to operate it is impaired to any degree by alcohol or a drug or by a combination of alcohol and a drug;”*

E Human Evaluation Datasheet

E.1 Paper and Supplementary Resources (Questions 1.1–1.3)

Question 1.1: Link to paper reporting the evaluation experiment. If the paper reports more than one experiment, state which experiment you’re completing this sheet for. Or, if applicable, enter ‘for preregistration.’

For preregistration.

Question 1.2: Link to website providing resources used in the evaluation experiment (e.g. system outputs, evaluation tools, etc.). If there isn’t one, enter ‘N/A’.

N/A.

Question 1.3: Name, affiliation and email address of person completing this sheet, and of contact author if different.

Will be completed upon acceptance.

E.2 System (Questions 2.1–2.5)

Question 2.1: What type of input do the evaluated system(s) take? Select all that apply. If none match, select ‘Other’ and describe.

Check-box options (select all that apply):

- ✓ **raw/structured data:** numerical, symbolic, and other data, possibly structured into trees, graphs, graphical models, etc. May be the input e.g. to Referring Expression Generation (REG), end-to-end text generation, etc. NB: excludes linguistic structures.
- **deep linguistic representation (DLR):** any of a variety of deep, underspecified, semantic representations, such as abstract meaning representations (AMRs; Banarescu et al., 2013) or discourse representation structures (DRSs; ?).
- **shallow linguistic representation (SLR):** any of a variety of shallow, syntactic representations, e.g. Universal Dependency (UD) structures; typically the input to surface realisation.

- **text: subsentential unit of text:** a unit of text shorter than a sentence, e.g. Referring Expressions (REs), verb phrase, text fragment of any length; includes titles/headlines.
- **text: sentence:** a single sentence (or set of sentences).
- **text: multiple sentences:** a sequence of multiple sentences, without any document structure (or a set of such sequences).
- **text: document:** a text with document structure, such as a title, paragraph breaks or sections, e.g. a set of news reports for summarisation.
- **text: dialogue:** a dialogue of any length, excluding a single turn which would come under one of the other text types.
- **text: other:** input is text but doesn’t match any of the above *text:** categories.
- **speech:** a recording of speech.
- **visual:** an image or video.
- **multi-modal:** catch-all value for any combination of data and/or linguistic representation and/or visual data etc.
- **control feature:** a feature or parameter specifically present to control a property of the output text, e.g. positive stance, formality, author style.
- **no input (human generation):** human generation⁷, therefore no system inputs.
- **other (please specify):** if input is none of the above, choose this option and describe it.

Question 2.2: What type of output do the evaluated system(s) generate? Select all that apply. If none match, select ‘Other’ and describe.

Check-box options (select all that apply):

- **raw/structured data:** numerical, symbolic, and other data, possibly structured into trees, graphs, graphical models, etc. May be the input e.g. to Referring Expression Generation (REG), end-to-end text generation, etc. NB: excludes linguistic structures.

⁷We use the term ‘human generation’ where the items being evaluated have been created manually, rather than generated by an automatic system.

- deep linguistic representation (DLR)**: any of a variety of deep, underspecified, semantic representations, such as abstract meaning representations (AMRs; Banarescu et al., 2013) or discourse representation structures (DRSs; ?).
 - shallow linguistic representation (SLR)**: any of a variety of shallow, syntactic representations, e.g. Universal Dependency (UD) structures; typically the input to surface realisation.
 - text: subsentential unit of text**: a unit of text shorter than a sentence, e.g. Referring Expressions (REs), verb phrase, text fragment of any length; includes titles/headlines.
 - ✓ **text: sentence**: a single sentence (or set of sentences).
 - text: multiple sentences**: a sequence of multiple sentences, without any document structure (or a set of such sequences).
 - text: document**: a text with document structure, such as a title, paragraph breaks or sections, e.g. a set of news reports for summarisation.
 - text: dialogue**: a dialogue of any length, excluding a single turn which would come under one of the other text types.
 - text: other**: select if output is text but doesn't match any of the above *text:** categories.
 - speech**: a recording of speech.
 - visual**: an image or video.
 - multi-modal**: catch-all value for any combination of data and/or linguistic representation and/or visual data etc.
 - human-generated 'outputs'**: manually created stand-ins exemplifying outputs.
 - other (please specify)**: if output is none of the above, choose this option and describe it.
- generated text from a representation of possible content. This could be attribute selection for REG (without the surface realisation step). Note that the output here is not text.
 - content ordering/structuring**: assigning an order and/or structure to content to be included in generated text. Note that the output here is not text.
 - aggregation**: converting inputs (typically *deep linguistic representations* or *shallow linguistic representations*) in some way in order to reduce redundancy (e.g. representations for 'they like swimming', 'they like running' → representation for 'they like swimming and running').
 - referring expression generation**: generating *text* to refer to a given referent, typically represented in the input as a set of attributes or a linguistic representation.
 - lexicalisation**: associating (parts of) an input representation with specific lexical items to be used in their realisation.
 - ✓ **deep generation**: one-step text generation from *raw/structured data* or *deep linguistic representations*. One-step means that no intermediate representations are passed from one independently run module to another.
 - surface realisation (SLR to text)**: one-step text generation from *shallow linguistic representations*. One-step means that no intermediate representations are passed from one independently run module to another.
 - feature-controlled text generation**: generation of text that varies along specific dimensions where the variation is controlled via *control features* specified as part of the input. Input is a non-textual representation (for feature-controlled text-to-text generation select the matching text-to-text task).
 - ✓ **data-to-text generation**: generation from *raw/structured data* which may or may not include some amount of content selection as part of the generation process. Output is likely to be *text:** or *multi-modal*.
 - dialogue turn generation**: generating a dialogue turn (can be a greeting or closing) from a representation of dialogue state and/or last turn(s), etc.
 - question generation**: generation of questions from given input text and/or knowledge base

Question 2.3: How would you describe the task that the evaluated system(s) perform in mapping the inputs in Q2.1 to the outputs in Q2.2? Occasionally, more than one of the options below may apply. If none match, select 'Other' and describe.

Check-box options (select all that apply):

- content selection/determination**: selecting the specific content that will be expressed in the

such that the question can be answered from the input.

- **question answering:** input is a question plus optionally a set of reference texts and/or knowledge base, and the output is the answer to the question.
- ✓ **paraphrasing/lossless simplification:** text-to-text generation where the aim is to preserve the meaning of the input while changing its wording. This can include the aim of changing the text on a given dimension, e.g. making it simpler, changing its stance or sentiment, etc., which may be controllable via input features. Note that this task type includes meaning-preserving text simplification (non-meaning preserving simplification comes under *compression/lossy simplification* below).
- **compression/lossy simplification:** text-to-text generation that has the aim to generate a shorter, or shorter and simpler, version of the input text. This will normally affect meaning to some extent, but as a side effect, rather than the primary aim, as is the case in *summarisation*.
- **machine translation:** translating text in a source language to text in a target language while maximally preserving the meaning.
- **summarisation (text-to-text):** output is an extractive or abstractive summary of the important/relevant/salient content of the input document(s).
- **end-to-end text generation:** use this option if the single system task corresponds to more than one of tasks above, implemented either as separate modules pipelined together, or as one-step generation, other than *deep generation* and *surface realisation*.
- **image/video description:** input includes *visual*, and the output describes it in some way.
- **post-editing/correction:** system edits and/or corrects the input text (typically itself the textual output from another system) to yield an improved version of the text.
- **other (please specify):** if task is none of the above, choose this option and describe it.

Question 2.4: Input Language(s), or ‘N/A’.

French.

Question 2.5: Output Language(s), or ‘N/A’.

French.

E.3 Output Sample, Evaluators, Experimental Design

E.3.1 Sample of system outputs (or human-authored stand-ins) evaluated (Questions 3.1.1–3.1.3)

Question 3.1.1: How many system outputs (or other evaluation items) are evaluated per system in the evaluation experiment? Answer should be an integer.

89.

Question 3.1.2: How are system outputs (or other evaluation items) selected for inclusion in the evaluation experiment? If none match, select ‘Other’ and describe.

Multiple-choice options (select one):

- **by an automatic random process from a larger set:** outputs were selected for inclusion in the experiment by a script using a pseudo-random number generator; don’t use this option if the script selects every *n*th output (which is not random).
- **by an automatic random process but using stratified sampling over given properties:** use this option if selection was by a random script as above, but with added constraints ensuring that the sample is representative of the set of outputs it was selected from, in terms of given properties, such as sentence length, positive/negative stance, etc.
- **by manual, arbitrary selection:** output sample was selected by hand, or automatically from a manually compiled list, without a specific selection criterion.
- ✓ **by manual selection aimed at achieving balance or variety relative to given properties:** selection by hand as above, but with specific selection criteria, e.g. same number of outputs from each time period.

- **Other (please specify):** if selection method is none of the above, choose this option and describe it.

Question 3.1.3: What is the statistical power of the sample size?

Following the methodology of Card et al. (2020), we obtained a statistical power of 1.0 on the output sample w.r.t the automatic evaluation metrics, the two best performing models (BARThez and *Criminel*/BART). We used their online script to estimate the statistical power.

E.3.2 Evaluators (Questions 3.2.1–3.2.4)

Question 3.2.1: How many evaluators are there in this experiment? Answer should be an integer.

Three.

Question 3.2.2: What kind of evaluators are in this experiment? Select all that apply. If none match, select ‘Other’ and describe. In all cases, provide details in the text box under ‘Other’.

Check-box options (select all that apply):

- ✓ **experts:** participants are considered domain experts, e.g. meteorologists evaluating a weather forecast generator, or nurses evaluating an ICU report generator.
- **non-experts:** participants are not domain experts.
- ✓ **paid (including non-monetary compensation such as course credits):** participants were given some form of compensation for their participation, including vouchers, course credits, and reimbursement for travel unless based on receipts.
- **not paid:** participants were not given compensation of any kind.
- **previously known to authors:** (one of the) researchers running the experiment knew some or all of the participants before recruiting them for the experiment.

- ✓ **not previously known to authors:** none of the researchers running the experiment knew any of the participants before recruiting them for the experiment.

- ✓ **evaluators include one or more of the authors:** one or more researchers running the experiment was among the participants.

- **evaluators do not include any of the authors:** none of the researchers running the experiment were among the participants.

- **Other (fewer than 4 of the above apply):** we believe you should be able to tick 4 options of the above. If that’s not the case, use this box to explain.

Question 3.2.3: How are evaluators recruited?

Evaluators (excluding one or more of the authors) were recruited by word of mouth, and have been interviewed prior to conduct the experiment.

Question 3.2.4: What training and/or practice are evaluators given before starting on the evaluation itself?

First, the evaluators have been introduced to the task of data-to-text generation. They then have been introduced to the dataset under study. They learned from an annotation guideline and have practiced on 5 examples before conducting the whole experiment. Evaluators did not need legal training since they had background knowledge on the domain.

Question 3.2.5: What other characteristics do the evaluators have, known either because these were qualifying criteria, or from information gathered as part of the evaluation?

Evaluators have been selected based on their educational level (2 years in law school) and their interest in criminal law.

E.3.3 Experimental design (Questions 3.3.1–3.3.8)

Question 3.3.1: Has the experimental design been preregistered? If yes, on which registry?

No.

Question 3.3.2: How are responses collected? E.g. paper forms, online survey tool, etc.

The answers were collected using a customized version of Prodigy⁸, hosted on Amazon Web Services.

Question 3.3.3: What quality assurance methods are used? Select all that apply. If none match, select ‘Other’ and describe. In all cases, provide details in the text box under ‘Other’.

Check-box options (select all that apply):

- ✓ *evaluators are required to be native speakers of the language they evaluate*: mechanisms are in place to ensure all participants are native speakers of the language they evaluate.
- automatic quality checking methods are used during/post evaluation*: evaluations are checked for quality by automatic scripts during or after evaluations, e.g. evaluators are given known bad/good outputs to check they’re given bad/good scores on MTurk.
- ✓ *manual quality checking methods are used during/post evaluation*: evaluations are checked for quality by a manual process during or after evaluations, e.g. scores assigned by evaluators are monitored by researchers conducting the experiment.
- evaluators are excluded if they fail quality checks (often or badly enough)*: there are conditions under which evaluations produced by participants are not included in the final results due to quality issues.

⁸<https://prodi.gy/>

- some evaluations are excluded because of failed quality checks*: there are conditions under which some (but not all) of the evaluations produced by some participants are not included in the final results due to quality issues.
- none of the above*: tick this box if none of the above apply.
- Other (please specify)*: use this box to describe any other quality assurance methods used during or after evaluations, and to provide additional details for any of the options selected above.

Question 3.3.4: What do evaluators see when carrying out evaluations? Link to screenshot(s) and/or describe the evaluation interface(s).

When carrying out evaluations, evaluators see the input data as well as three generations from three different models. They do not know which generation corresponds to which model. They then provide a score for each generation independently.

3.3.5: How free are evaluators regarding when and how quickly to carry out evaluations? Select all that apply. In all cases, provide details in the text box under ‘Other’.

Check-box options (select all that apply):

- evaluators have to complete each individual assessment within a set time*: evaluators are timed while carrying out each assessment and cannot complete the assessment once time has run out.
- evaluators have to complete the whole evaluation in one sitting*: partial progress cannot be saved and the evaluation returned to on a later occasion.
- ✓ *neither of the above*: Choose this option if neither of the above are the case in the experiment.
- Other (please specify)*: Use this space to describe any other way in which time taken or number of sessions used by evaluators is controlled in the experiment, and to provide additional details for any of the options selected above.

3.3.6: Are evaluators told they can ask questions about the evaluation and/or provide feedback? Select all that apply. In all cases, provide details in the text box under ‘Other’.

Check-box options (select all that apply):

- ✓ *evaluators are told they can ask any questions during/after receiving initial training/instructions, and before the start of the evaluation*: evaluators are told explicitly that they can ask questions about the evaluation experiment *before* starting on their assessments, either during or after training.
- evaluators are told they can ask any questions during the evaluation*: evaluators are told explicitly that they can ask questions about the evaluation experiment *during* their assessments.
- evaluators are asked for feedback and/or comments after the evaluation, e.g. via an exit questionnaire or a comment box*: evaluators are explicitly asked to provide feedback and/or comments about the experiment *after* their assessments, either verbally or in written form.
- None of the above*: Choose this option if none of the above are the case in the experiment.
- Other (please specify)*: use this space to describe any other ways you provide for evaluators to ask questions or provide feedback.

3.3.7: What are the experimental conditions in which evaluators carry out the evaluations? If none match, select ‘Other’ and describe.

Multiple-choice options (select one):

- ✓ *evaluation carried out by evaluators at a place of their own choosing, e.g. online, using a paper form, etc.*: evaluators are given access to the tool or form specified in Question 3.3.2, and subsequently choose where to carry out their evaluations.
- evaluation carried out in a lab, and conditions are the same for each evaluator*: evaluations are carried out in a lab, and conditions in which evaluations are carried out *are* controlled to be the same, i.e. the different evaluators all carry out the evaluations in identical conditions of quietness, same type of computer, same room, etc. Note we’re not after very fine-grained differences here, such as time of day or temperature, but the line is difficult to draw, so some judgment is involved here.
- evaluation carried out in a lab, and conditions vary for different evaluators*: choose this option if evaluations are carried out in a lab, but the preceding option does not apply, i.e. conditions in which evaluations are carried out are *not* controlled to be the same.
- evaluation carried out in a real-life situation, and conditions are the same for each evaluator*: evaluations are carried out in a real-life situation, i.e. one that would occur whether or not the evaluation was carried out (e.g. evaluating a dialogue system deployed in a live chat function on a website), and conditions in which evaluations are carried out *are* controlled to be the same.
- evaluation carried out in a real-life situation, and conditions vary for different evaluators*: choose this option if evaluations are carried out in a real-life situation, but the preceding option does not apply, i.e. conditions in which evaluations are carried out are *not* controlled to be the same.
- evaluation carried out outside of the lab, in a situation designed to resemble a real-life situation, and conditions are the same for each evaluator*: evaluations are carried out outside of the lab, in a situation intentionally similar to a real-life situation (but not actually a real-life situation), e.g. user-testing a navigation system where the destination is part of the evaluation design, rather than chosen by the user. Conditions in which evaluations are carried out *are* controlled to be the same.
- evaluation carried out outside of the lab, in a situation designed to resemble a real-life situation, and conditions vary for different evaluators*: choose this option if evaluations are carried out outside of the lab, in a situation intentionally similar to a real-life situation, but the preceding option does not apply, i.e. conditions in which evaluations are carried out are *not* controlled to be the same.
- Other (please specify)*: Use this space to pro-

vide additional, or alternative, information about the conditions in which evaluators carry out assessments, not covered by the options above.

3.3.8: Unless the evaluation is carried out at a place of the evaluators' own choosing, briefly describe the (range of different) conditions in which evaluators carry out the evaluations.

N/A.

E.4 Quality Criterion n – Definition and Operationalisation

E.4.1 Quality criterion properties (Questions 4.1.1–4.1.3)

Question 4.1.1: What type of quality is assessed by the quality criterion?

Multiple-choice options (select one):

- ✓ **Correctness:** select this option if it is possible to state, generally for all outputs, the conditions under which outputs are maximally correct (hence of maximal quality). E.g. for Grammaticality, outputs are (maximally) correct if they contain no grammatical errors; for Semantic Completeness, outputs are correct if they express all the content in the input.
- **Goodness:** select this option if, in contrast to correctness criteria, there is no single, general mechanism for deciding when outputs are maximally good, only for deciding for two outputs which is better and which is worse. E.g. for Fluency, even if outputs contain no disfluencies, there may be other ways in which any given output could be more fluent.
- **Features:** choose this option if, in terms of property X captured by the criterion, outputs are not generally better if they are more X , but instead, depending on evaluation context, more X may be better or less X may be better. E.g. outputs can be more specific or less specific, but it's not the case that outputs are, in the general case, better when they are more specific.

Question 4.1.2: Which aspect of system outputs is assessed by the quality criterion?

Multiple-choice options (select one):

- **Form of output:** choose this option if the criterion assesses the form of outputs alone, e.g. Grammaticality is only about the form, a sentence can be grammatical yet be wrong or nonsensical in terms of content.
- ✓ **Content of output:** choose this option if the criterion assesses the content/meaning of the output alone, e.g. Meaning Preservation only assesses output content; two sentences can be considered to have the same meaning, but differ in form.
- **Both form and content of output:** choose this option if the criterion assesses outputs as a whole, not just form or just content. E.g. Coherence is a property of outputs as a whole, either form or meaning can detract from it.

Question 4.1.3: Is each output assessed for quality in its own right, or with reference to a system-internal or external frame of reference?

Multiple-choice options (select one):

- **Quality of output in its own right:** choose this option if output quality is assessed without referring to anything other than the output itself, i.e. no system-internal or external frame of reference. E.g. Poeticness is assessed by considering (just) the output and how poetic it is.
- ✓ **Quality of output relative to the input:** choose this option if output quality is assessed relative to the input. E.g. Answerability is the degree to which the output question can be answered from information in the input.
- **Quality of output relative to a system-external frame of reference:** choose this option if output quality is assessed with reference to system-external information, such as a knowledge base, a person's individual writing style, or the performance of an embedding system. E.g. Factual Accuracy assesses outputs relative to a source of real-world knowledge.

E.4.2 Evaluation mode properties (Questions 4.2.1–4.2.3)

Questions 4.2.1–4.2.3 record properties that are orthogonal to quality criteria, i.e. any given quality criterion can in principle be combined with any of the modes (although some combinations are more common than others).

Question 4.2.1: Does an individual assessment involve an objective or a subjective judgment?

Multiple-choice options (select one):

- ✓ **Objective:** Examples of objective assessment include any automatically counted or otherwise quantified measurements such as mouse-clicks, occurrences in text, etc. Repeated assessments of the same output with an objective-mode evaluation method always yield the same score/result.
- **Subjective:** Subjective assessments involve ratings, opinions and preferences by evaluators. Some criteria lend themselves more readily to subjective assessments, e.g. Friendliness of a conversational agent, but an objective measure e.g. based on lexical markers is also conceivable.

Question 4.2.2: Are outputs assessed in absolute or relative terms?

Multiple-choice options (select one):

- **Absolute:** choose this option if evaluators are shown outputs from a single system during each individual assessment.
- ✓ **Relative:** choose this option if evaluators are shown outputs from multiple systems at the same time during assessments, typically ranking or preference-judging them.

Question 4.2.3: Is the evaluation intrinsic or extrinsic?

Multiple-choice options (select one):

- **Intrinsic:** Choose this option if quality of outputs is assessed *without* considering their *effect* on something external to the system, e.g. the performance of an embedding system or of a user at a task.
- ✓ **Extrinsic:** Choose this option if quality of outputs is assessed in terms of their *effect* on something external to the system such as the performance of an embedding system or of a user at a task.

E.4.3 Response elicitation (Questions 4.3.1–4.3.11)

Question 4.3.1: What do you call the quality criterion in explanations/interfaces to evaluators? Enter 'N/A' if criterion not named.

Legal accuracy.

Question 4.3.2: What definition do you give for the quality criterion in explanations/interfaces to evaluators? Enter 'N/A' if no definition given.

We define legal accuracy as being a text that respectfully captures the input data w.r.t the criminal code, the plea and the verdict. In most cases, legal accuracy w.r.t the criminal code is the hardest part of the task for neural networks.

Question 4.3.3: Size of scale or other rating instrument (i.e. how many different possible values there are). Answer should be an integer or 'continuous' (if it's not possible to state how many possible responses there are). Enter 'N/A' if there is no rating instrument.

10.

Question 4.3.4: List or range of possible values of the scale or other rating instrument. Enter 'N/A', if there is no rating instrument.

1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

Question 4.3.5: How is the scale or other rating instrument presented to evaluators? If none match, select ‘Other’ and describe.

Multiple-choice options (select one):

- **Multiple-choice options:** choose this option if evaluators select exactly one of multiple options.
- **Check-boxes:** choose this option if evaluators select any number of options from multiple given options.
- **Slider:** choose this option if evaluators move a pointer on a slider scale to the position corresponding to their assessment.
- **N/A (there is no rating instrument):** choose this option if there is no rating instrument.
- ✓ **Other (please specify):** choose this option if there is a rating instrument, but none of the above adequately describe the way you present it to evaluators. Use the text box to describe the rating instrument and link to a screenshot.

Due to the limitations of Prodigy regarding their slider component (only one per page), we used a free-form text box. Since we have few, highly skilled evaluators, it was not a problem collecting data.

Question 4.3.6: If there is no rating instrument, describe briefly what task the evaluators perform (e.g. ranking multiple outputs, finding information, playing a game, etc.), and what information is recorded. Enter ‘N/A’ if there is a rating instrument.

N/A.

Question 4.3.7: What is the verbatim question, prompt or instruction given to evaluators (visible to them during each individual assessment)?

Do subsequent generations capture the data from the docket file? Rate on a scale of 1 to 10.

Question 4.3.8: Form of response elicitation. If none match, select ‘Other’ and describe.

*Multiple-choice options (select one):*⁹

- **(dis)agreement with quality statement:** Participants specify the degree to which they agree with a given quality statement by indicating their agreement on a rating instrument. The rating instrument is labelled with degrees of agreement and can additionally have numerical labels. E.g. *This text is fluent — 1=strongly disagree...5=strongly agree.*
- **direct quality estimation:** Participants are asked to provide a rating using a rating instrument, which typically (but not always) mentions the quality criterion explicitly. E.g. *How fluent is this text? — 1=not at all fluent...5=very fluent.*
- **relative quality estimation (including ranking):** Participants evaluate two or more items in terms of which is better. E.g. *Rank these texts in terms of fluency; Which of these texts is more fluent?; Which of these items do you prefer?.*
- ✓ **counting occurrences in text:** Evaluators are asked to count how many times some type of phenomenon occurs, e.g. the number of facts contained in the output that are inconsistent with the input.
- **qualitative feedback (e.g. via comments entered in a text box):** Typically, these are responses to open-ended questions in a survey or interview.
- **evaluation through post-editing/annotation:** Choose this option if the evaluators’ task consists of editing or inserting annotations in text. E.g. evaluators may perform error correction and edits are then automatically measured to yield a numerical score.
- **output classification or labelling:** Choose this option if evaluators assign outputs to categories. E.g. *What is the overall sentiment of this piece of text? — Positive/neutral/negative.*
- **user-text interaction measurements:** choose this option if participants in the evaluation experiment interact with a text in some way, and

⁹Explanations adapted from Howcroft et al. (2020).

measurements are taken of their interaction. E.g. reading speed, eye movement tracking, comprehension questions, etc. Excludes situations where participants are given a task to solve and their performance is measured which comes under the next option.

- **task performance measurements:** choose this option if participants in the evaluation experiment are given a task to perform, and measurements are taken of their performance at the task. E.g. task is finding information, and task performance measurement is task completion speed and success rate.
- **user-system interaction measurements:** choose this option if participants in the evaluation experiment interact with a system in some way, while measurements are taken of their interaction. E.g. duration of interaction, hyperlinks followed, number of likes, or completed sales.
- **Other (please specify):** Use the text box to describe the form of response elicitation used in assessing the quality criterion if it doesn't fall in any of the above categories.

Question 4.3.9: How are raw responses from participants aggregated or otherwise processed to obtain reported scores for this quality criterion? State if no scores reported.

Macro-averages are computed from numerical scores to provide summary, per-system results.

Question 4.3.10: Method(s) used for determining effect size and significance of findings for this quality criterion.

What to enter in the text box: A list of methods used for calculating the effect size and significance of any results, both as reported in the paper given in Question 1.1, for this quality criterion. If none calculated, state 'None'. None.

Question 4.3.11: Has the inter-annotator and intra-annotator agreement between evaluators for this quality criterion been measured? If yes, what method was used, and what are the agreement scores?

Krippendorff's alpha is used to measure inter-annotator agreement. Krippendorff's alpha is of 0.84.

F Ethics

Question 5.1: Has the evaluation experiment this sheet is being completed for, or the larger study it is part of, been approved by a research ethics committee? If yes, which research ethics committee?

No.

Question 5.2: Do any of the system outputs (or human-authored stand-ins) evaluated, or do any of the responses collected, in the experiment contain personal data (as defined in GDPR Art. 4, §1: <https://gdpr.eu/article-4-definitions/>)? If yes, describe data and state how addressed.

No.

Question 5.3: Do any of the system outputs (or human-authored stand-ins) evaluated, or do any of the responses collected, in the experiment contain special category information (as defined in GDPR Art. 9, §1: <https://gdpr.eu/article-9-processing-special-categories-of-personal-data-prohibited/>)? If yes, describe data and state how addressed.

No.

Question 5.4: Have any impact assessments been carried out for the evaluation experiment, and/or any data collected/evaluated in connection with it? If yes, summarise approach(es) and outcomes.

No.

Automatic Generation of Factual News Headlines in Finnish

Maximilian Koppatz¹, Khalid Alnajjar¹, Mika Hämäläinen¹ and Thierry Poibeau²

¹ University of Helsinki

² École Normale Supérieure-PSL and CNRS and Université Sorbonne nouvelle

firstname.lastname@helsinki.fi

Abstract

We present a novel approach to generating news headlines in Finnish for a given news story. We model this as a summarization task where a model is given a news article, and its task is to produce a concise headline describing the main topic of the article. Because there are no openly available GPT-2 models for Finnish, we will first build such a model using several corpora. The model is then fine-tuned for the headline generation task using a massive news corpus. The system is evaluated by 3 expert journalists working in a Finnish media house. The results showcase the usability of the presented approach as a headline suggestion tool to facilitate the news production process.

1 Introduction

Authoring a good headline is an essential step in the process of writing and publishing news articles. A good headline should be an apt and concise description of the contents of the article. It should also be captivating so that it makes a potential reader interested in reading the article in addition to following the guidelines set by the news agency in question. Good and bad headlines have also a great impact on the number of visitors (Dor, 2003; Kuiken et al., 2017), which directly translates into revenue on ad supported news websites.

It is very typical to use A/B testing to study which headline candidates are more successful in engaging users. This testing requires there to be headline candidates to test to begin with. For this reason, editors need to write multiple headline candidates for a single news article. This task takes a lot of time away from other editorial work especially since the people inventing the headlines are very often not the same people who write the articles. According to journalists working at the Finnish press house Sanoma, editors often times invent dozens of alternative headlines a day for news articles. Needless to say there is a commercial interest in automating this task.

It is not straightforward to automatically generate news headlines that are useful. A usable headline is expected to convey correct facts and be thematically relevant. At the same time, there must be diversity in the generated headlines given that the press houses want to have access to multiple headline variants. There is a communicative-creative trade off (see Hämäläinen and Honkela (2019)) in how creative the system can be while still conveying the desired factual meaning. Additionally, the generated headlines must be interesting, because no reader would read a news story that sound boring from the very beginning.

In this paper, we represent a method for conditional headline generation by using a generative autoregressive Transformer (Vaswani et al., 2017) based language model that is fine-tuned for the headline generation task in Finnish. The approach we follow can be seen as a special case of text summarization. Instead of the target being a full summary, it is a very compact headline. The reason for approaching the problem for this angle instead of resorting to a masked language model such as BERT (Devlin et al., 2018), is that training autoregressive language models is computationally easier and faster than masked language models. This is because masked language models are trained to predict only a small percentage of words in a text during each forward pass, while autoregressive language models predict every word during every pass.

The main contributions of this paper are the following:

- We train the first Finnish GPT-2 (Radford et al., 2019) model
- We fine-tune the model for the downstream task of headline generation for the morphologically rich Finnish language
- We present a human evaluation using real journalists who invent headlines as their day job

2 Related Work

In contemporaneous studies, neural headline generation is approached from the point of view of text summarization. Text summarization in itself has traditionally been divided into extractive and abstractive summarization. Currently, both of these types of tasks are tackled with approaches that utilize the Transformer (Song et al., 2020; Bukhtiyarov and Gusev, 2020; Liu and Lapata, 2019).

A common type of an approach for both extractive and abstractive summarization is an encoder-decoder type language model such as BertSumExt (Liu and Lapata, 2019) and PEGASUS (Zhang et al., 2020). Summarization as a seq2seq problem suits the encoder-decoder model-paradigm well as you have a source and a target text like in NMT problems. In this setup, abstractive summarization is performed by the generative decoder part. For purely extractive tasks, the decoder is often replaced by some form of classifier which selects which tokens in the input should be in the resulting summary.

Another type of an approach, and the one used in this paper, is to fine-tune a GPT-2 (Radford et al., 2019) style auto-regressive language model for the summarization task (Kieuvongngam et al., 2020; Song et al., 2020). These approaches perform some form of concatenation of the target summary to the end of the source text with special tokens as delimiters between source and target. This approach does not fit the paradigm of sequence transduction as well as encoder-decoder setups, but does have its advantages. For one, all of the parameters are reused maximally, as the entire model network is pre-trained on text generation. When fine-tuning for the summary, this continues to be the case. Encoder-decoder setups tend to become more complex.

Recent headline generation approaches tend to use some form of BLEU or ROUGE for automatic evaluation of the models (Matsumaru et al., 2020; Bukhtiyarov and Gusev, 2020; Tilk and Alumäe, 2017). These metrics are naturally used for summarization as well. Furthermore, Beam Search is a common way to perform the generation.

For Finnish in particular, the literature for neural headline generation and summarization is scarce. Currently however, most work regarding Finnish headline and text generation seems to be using more conventional NLP, rule-based and statistical methods (Leppänen et al., 2017; Hämäläinen and

Alnajjar, 2019; Hämäläinen and Rueter, 2018).

For news focused Finnish NLP there has been work on generating sports reports from event data using a pointer-generation network (Kanerva et al., 2019), generation of creative headlines in Finnish using templates (Alnajjar et al., 2019) and rumor detection in Finnish news (Hämäläinen et al., 2021) using BERT and LSTMs.

How creative NLG systems are evaluated has been investigated in recent work (Hämäläinen and Alnajjar, 2021b), and the evaluation done in this paper roughly follows the conclusions drawn. Specifically, to evaluate the generation of the model aligned with what task the model was designed and trained to perform (Hämäläinen and Alnajjar, 2021a). Following this idea, the evaluation of the model in this paper is not relying simply on offline metrics, but on manual structured review by domain-experts with criteria relevant to the real-world use case.

3 Data

This section details the data, filtering, processing and tokenization used in this paper. There are two separate modeling tasks we perform: unsupervised generative pre-training and generative fine-tuning. For this reason, we make sure that there are always two columns in the dataset: "body" and "title". The body column contains everything except headlines and is used as the pre-training data. Later, the headlines are added for the fine-tuning task.

3.1 Corpora

The data used for pre-training the language model consists of four corpora concatenated together: Sanoma, Wikipedia, Yle and Ylilauta.

The Sanoma corpus is our primary and largest corpus. It is a proprietary corpus of news articles from the most important Finnish news paper *Helsingin Sanomat* and the widely spread yellow press paper *Ilta-Sanomat*. This corpus contains approximately 3.8 million Sanoma news articles published between the year 1990 and 2021. The topic coverage is as broad as one would expect from news media, ranging from domestic and international politics to sports and culture events.

The Sanoma corpus contains the headline, ingress, and article body for most articles. We concatenate the ingress to the body text with double newlines between. This data was saved into parquet format with "title" and "body" columns.

Headlines are kept separate because they are used only in the fine-tuning phase. This holds true to all corpora with headlines.

Wikipedia is a great corpus for language modeling, as it is freely available and contains information about the world. The corpus contains pages containing information about countries, people, history, science and much more. This is particularly useful for unsupervised language model pre-training as the model can learn from the information found in Wikipedia. The Finnish Wikipedia dump¹ from 24.11.2020 was used. This dump was parsed into a parquet file with again a "title" and "body" column. The dump contains 463,780 pages.

A corpus of news articles from **Yle**² was parsed into the same "title" and "body" format as the Sanoma and Wikipedia corpora. This corpus is small and only contributed around 100 000 articles.

The **Ylilauta** corpus³ contains 335,004 messages from the Ylilauta forums. These messages are quite different to the rest of the data used, as this is not structured text. This text is also colloquial. Furthermore, this corpus does not contain headlines. As it represents a different textual domain, it makes it possible for the model to learn a representation of colloquial Finnish as well.

3.2 Tokenizer

The tokenization procedure must be able to tokenize any text string into tokens that all exist in the vocabulary of the language model. *Byte-pair-encodings* (BPE) (Sennrich et al., 2016), and variations of it, is a common way to tokenize text for transformer language models especially for NMT. BPE strikes a balance between word-level and character-level tokenization by using subword-tokenization. It is able to express almost any string, like character-level tokenization, but without needing to treat each character separately which would result in very long sequences.

For the model, the number of merges was set to have a resulting total vocabulary size of 50,000, which is close in size to the GPT-2 vocabulary. The Byte-level BPE vocabulary was learned on the entire corpus. Additionally, included into this vocabulary are some special tokens which we added: `<sos>`, `<eos>` for start and end of text tokens, `<unk>` for unknown tokens just in case there is an error, `<special1>`, `<special2>`, `<special3>` tokens

¹<https://dumps.wikimedia.org/finwiki/latest/>

²<http://urn.fi/urn:nbn:fi:lb-2017070501>

³<http://urn.fi/urn:nbn:fi:lb-2016101210>

reserved for possible of later downstream use when fine-tuning the model for a specific task. The special tokens never appear in the pre-training corpus, and `<special1>` is used later on when fine-tuning to generate headlines.

4 Building a Finnish GPT-2

Our approach to creating a headline generating model is based on fine-tuning a language model learned by unsupervised generative pre-training. As such a model does not exist for Finnish, we have to train one.

4.1 Model Specifications

The language model in this paper are decoder Transformers, with a few key modifications. The modifications closely follow those made to GPT-2 as compared to the original Transformer.

Positioning of the layer normalization has been to follow GPT-2. Originally layer normalization was applied after the residual connections. This was modified by moving layer normalization to the input of each sublayer, and adding an additional layer normalization to the output of the final self-attention layer.

Positional embeddings are learned instead of sinusoidal. The reason for this is that BERT and the GPT variants use learned positional embeddings as well. This involves adding another embedding matrix to the neural architecture in addition to the token embedding matrix. The difference is that the position embedding matrix keys are the position integers of a token relative to the text it resides in, while the token embedding matrix has the vocabulary id of the token as the index.

Again following GPT-2, the network parameters are initialized by sampling from $N(0, \frac{0.02}{\sqrt{n}})$, where n is the number of residual layers. From our experiments, this change is crucial for the convergence of larger model sizes. The model sized discussed in this paper which are of size L and larger did not converge at all without this change.

Like GPT-2, Gaussian Error Linear Unit (GELU) (Hendrycks and Gimpel, 2016) was used as the activation function in the network instead of ReLU (Agarap, 2018). We used the AdamW (Loshchilov and Hutter, 2017) optimizer with a learning rate α . For the final model, $d_{model} = 1280$ and $n_{warmup} = 2000$. This was done by doing several restarts and observing when the gradients overflow and the training breaks down as the learning rate is

increased during warmup. The formula is tuned so that the peak learning rate is lower than the learning rate was when overflow was observed.

GPT-2 had additional L2 regularization which is omitted in this paper work. Finally, our models do not use dropout regularization. This is due to seeing better validation set convergence without it when testing hyperparameters on smaller test training runs. Since training the Transformer model takes time, omitting dropout regularization allows the model to converge slightly faster in terms of time. We ran several small-scale experiments with varying degrees of dropout and found that it did not significantly affect the end validation perplexity.

4.2 Results of Pre-training

Perplexity (Equation 1) is a commonly used measure of the performance of a language model (Radford and Narasimhan, 2018; Radford et al., 2019; Brown et al., 2020). The perplexity of a model given a text is calculated based on the probability assigned to each actual token in the text given its context. A convenient way to calculate it is by exponentiating the negative log likelihood loss:

$$ppl = e^{-\sum_{t=1}^n \log(p(x^{(t)} | x^{(1)}, \dots, x^{(t-1)}), \theta)} \quad (1)$$

A language model with a vocabulary of size V will have a perplexity score of exactly V if it always predicts the uniform distribution for each token. A random language model will average around V perplexity as well. This is because on average, the correct token in a text is given $\frac{1}{V}$ probability by such a LM. Conversely, a LM that always predicts the token correctly with 100% assigned probability will have a perplexity score of 1.

The resulting train and test perplexities achieved for the various models are found in Table 1. The medium and large models were trained for 4 epochs each, while the second large model was trained for less than 3 epochs. The training times were approximately 2 weeks for all models, and the medium and first large models were trained on half of the full corpus. This test reveals that in this case, the size of the training set has more impact on the resulting model perplexity than the size of the model. The implication is that increasing the size of the model won't increase the quality of the model significantly if the amount of training data is insufficient. Due to this and that it would have taken more than 2 weeks to train the XL sized model, we chose not to

Size	ppl_train	ppl_test
medium	17.9	22.9
large	17.4	21.6
large2	14.0	17.8

Table 1: Train and test perplexity scores.

train a full XL sized GPT model. Additionally, the generative performance from manual testing was good enough.

5 Headline Generation

This chapter describes how we tackle the task of conditional headline generation by using transfer learning. The final Finnish LM is used as the base model. we describe the training procedure and decoding algorithm design first, followed by the description of a domain-expert evaluation of the headline generative performance.

The final Finnish LM is loaded from its latest checkpoint, including both its parameters and optimizer state. Training is resumed but with changes to the learning rate, input structure, and validation generation. The loss calculation is altered as well, to focus the learning purely on the task of headline generation.

The training, validation and testing corpora are filtered, removing texts that do not have a headline or are not news articles. The texts are re-formatted, clipping the body of the text to the first 448 tokens. The special token $\langle special \rangle$, not previously shown to the model, is appended to the end of the clipped body text. The headline of the text is then appended following the special token, followed by the $\langle eos \rangle$ token signifying the end of the output. The idea is to learn a pattern where text that follows a special token is always a headline summarizing the preceding text, followed by the end token. Ideally then when the LM is given any text prompt ending in the special token, the output of applying a generation algorithm would be a relevant headline. The clipping procedure results in a portion of the news articles body text not to be completely shown to the model. This clipping must be done due to the model having a maximum context width of 512 tokens in order to fit the headlines at the end. We chose to keep the beginning of the texts due to news articles being structured in a way where the most important content tends to be written in the beginning of the article.

The data is no longer fed to the model in a dense

square matrix format as when pre-training the LM. That method would be separating the headline of an article and the tokens of the article itself much of the time into separate rows in the input tensors. Ideally the corpus would be sorted according to length and fed to the model in variable sized batches of instances with similar length. We omitted this step for convenience, as we only ran the fine-tuning runs for one epoch each.

Sampling based algorithms such as nucleus sampling don't seem to lend themselves well for the task of headline generation. This can be seen by simply observing the random nature of the headline generation when using sampling based methods, both from the validation output and from manual testing. The requirements for headlines are stricter than that of creative text continuation, in that the headline must at least summarize accurately the article, and not invent things not stated in the text.

Beam Search works better for this task. The problem with Beam Search for practical applications though, is that if you want several headlines, it produces the same headline with only slight variations. Often, with just one word differences at the end.

Diverse Beam Search (DBS) (Vijayakumar et al., 2016) is an alternative to BS which addresses the diversity issue by decoding in a doubly greedy manner, optimizing both the sequence probability under the model as well as the diversity.

At a high level, the B beams are divided into G groups. In addition, a similarity penalty is introduced. This penalizes subsequent groups sequence probability score by a similarity penalty term multiplied by λ , a parameter for similarity penalty strength. In this work, the similarity score is the integer number of times in previous beam groups the proposed token has been selected during this step.

In this algorithm we have G separate groups of beam-search. For each decoding step, the groups of beam-search are advanced in consecutive order. For each consecutive token in the decoding process, the first group has no diversity penalty from previous groups, and as such is simply beam search. For each consecutive group, regular beam search is conducted but with the sequences penalized when the proposed token has been used in previous groups during this step. This adds diversity between the groups already from the first token if λ is high enough, due to each group beginning the headline

with a different token.

Vanilla DBS does not address the repetitiveness problem. While repetitiveness seems to happen less when generating headlines, it still does happen that a name or sentence is repeated. For this reason, we added a second penalty which is beam-specific: λ_{repeat} . This is a penalty applied to the probability score of continuations to a sequence when the proposed token has previously been used in the sequence in question. This is mainly to prevent outputs such as "*Niinistö tapasi Niinistön*" (Niinistö met Niinistö). If λ_{repeat} is set too high, then grammar can suffer due to proper suffixes being penalized too harshly. One could say that if the model was good, this should be unnecessary. Unfortunately, it seems that this repetitive behaviour is common in this type of MLE language model optimization.

The likelihood under the model for a sequence in Beam Search in general is the joint probability of the sequence calculated using the Chain Rule of Probabilities by multiplying each token probability conditioned on the context together. In this case of generating headlines, this causes shorter headlines to have a higher probability. They are usually safer but more boring headlines. In order to combat this, we added a decay parameter β . This parameter is multiplied together with the current log-probability of the sequence so far before adding the log-probability of the proposed token to it. The result is equivalent for $\beta = 1$ and results in longer headlines when $\beta < 1$.

Our implementation of DBS has 6 parameters in total. G and B for the number of groups and number of beams per group. We selected $G = 4$ and $B = 2$ for 2 beams per group and 4 output headlines, totalling in 8 beams. The maximum length of a headline is another hyperparameter which we set to 48 tokens. The 3 remaining hyperparameters λ , λ_{repeat} and β were tuned algorithmically, because it was too much manual work with unclear results to tune these manually.

We used Gaussian Process optimization (Snoek et al., 2012) to select these 3 parameters. The objective function we used was the BLEU (Papineni et al., 2002) score of the generated set of headlines with regards to the true headline for 100 articles. We opted for GP optimization instead of grid search as grid search would have taken too long, as generating one set of headlines once already takes several seconds.

For λ we observed two separate points of interest where the target (BLEU) is at maxima. These are consequently the points with highest search density for this hyper-parameter. The final values for the hyper-parameters were $\lambda = 0.71$, $\lambda_{repeat} = 3$ and $\beta = 0.87$. Notably, 3 was the maximum we had set for λ_{repeat} , making higher values possibly better still.

6 Results and Evaluation

As previously mentioned, the evaluation of models should be conducted in a way that measures the performance of the actual desired task at hand. For this reason, calculating BLEU or similar on an offline corpus is not an accurate representation of the performance of the model when it comes to generating real world headlines. The question we seek to answer in this paper is how well can this model perform in real-world use in the newsroom as a tool to help editors headline articles.

6.1 Study Design

To thoroughly answer the research question, we generated a set of headlines for new articles, and had domain-experts evaluate them by hand according to three key criteria. We picked 100 random news articles from Helsingin Sanomat (HS) and Ilta-Sanomat (IS) not contained in the original corpus. For each article, we generated four headlines using our implementation of DBS and the optimized parameter set. We made an Excel worksheet⁴ where each article had its text in one column, and in another column its four generated headlines as well as the real original headline in a random position in the headline set. The worksheet has a column for each of the three criteria which the evaluators fill with 1 for the headline passing the criterion and 0 otherwise. The criteria are in order of difficulty for the model to achieve, with the first criterion being the easiest and the third criterion being the most difficult. Additionally, passing a criterion means passing the preceding criteria as well. The criteria are language, usable and good.

Language If disregarding the article text, is the headline on its own correct Finnish? Does this headline make sense to a human being? We elected to have this criterion separate from the next one to get a better understanding of where and how the performance breaks down.

Usable Could this headline be used for the given text in the real world? Does it match the text in the news article without misquoting and without errors? This is the most important question in terms of how good this model is for real-world use.

Good Is this headline good enough for the editor to be comfortable publishing the article with it without feeling the need to edit it or come up with variants? This final criterion is a subjective one but we decided to keep it separate from the usable criterion as they are fundamentally different.

Additionally, there's an optional open feedback column, as well as summary open feedback at the end of filling in the excel.

Three editors, one from Helsingin Sanomat and two from Ilta-Sanomat volunteered to perform this evaluation. Each one has extensive experience in headlining articles, sometimes coming up with dozens of headlines in a day. The final answers for each question are selected as the majority vote of the three. It took two weeks for them to fill in their answers. The real headline was inserted randomly as a control for possible anti-machine bias and as a baseline reference (Charnley et al., 2012).

Out of the 500 headlines, 467 received an evaluation from all three evaluators. Some of the headlines were not evaluated due to the source text having been incorrectly parsed, leaving out names of people and places and was deemed by the evaluator(s) to be best left unanswered. Some headlines seem to have been simply forgotten. Most of the following tables have the metrics for the real and the generated headlines separate for baseline reference.

The acceptance percentages for each of the three evaluation criteria per individual evaluator are shown in Table 2. We can see that evaluator A seems to have been able to distinguish between the real and generated headlines better than the other two evaluators, while evaluator B was the most forgiving.

The inter-annotator agreement per criterion measured by Fleiss' kappa (Fleiss, 1971) is shown in Table 3. Fleiss' kappa represents the degree of agreement when accounting for agreement by chance based on the ratio of passing versus rejecting the criteria. A positive number between 0 and 1 means there is more agreement than by chance, while a negative number between 0 and -1 indicates more disagreement than by chance.

For real headlines the degree of agreement is

⁴<https://zenodo.org/record/5985728>

Language			
Evaluator	A	B	C
Real	1.0	0.97	0.785
Generated	0.79	0.90	0.775

Usable			Good		
A	B	C	A	B	C
0.91	0.80	0.77	0.84	0.76	0.47
0.22	0.43	0.37	0.13	0.40	0.20

Table 2: The response acceptance ratio for each evaluator separately for Language, Usable and Good criteria separated by real headlines and generated headlines.

negative and close to chance. This is expected, as the majority of real headlines pass the criteria and the criteria are inherently slightly subjective.

The agreement in the three criteria for the generated headlines was modest but clearly greater than chance. The merely modest inter-annotator agreement shows numerically how the generated headlines often have errors that are hard to detect, as clearer errors would yield a high degree of agreement. The goodness criterion has the lowest inter-annotator agreement despite the model failing this criterion the most, as it is the most subjective.

Type	Language	Usable	Good
Real	-0.09	-0.02	-0.07
Generated	0.35	0.38	0.30

Table 3: Inter-annotator agreement measured by Fleiss’ kappa.

The headlines performed equally well per brand, as seen in Table 4. The language criterion was the only criterion where there was a notable difference between the brands. The model seems to have a slightly easier time with HS articles.

Brand	Language	Usable	Good
HS	0.91	0.31	0.20
IS	0.82	0.30	0.21

Table 4: Acceptance rates by brand. Both brands had approximately the same amount of headlines.

The final result of the survey where the headlines are scored for each criteria according to a majority vote is shown in Table 5. A headline passes a criterion if at least two out of the three evaluators vote to pass. we have the real control headlines

separate from the generated headlines as a baseline reference. Additionally, these tables show both the total acceptance rates as well as acceptance rates for headlines that have passed the preceding criteria. We can see that the language criterion is where the model performs by far the best as expected. The performance breakdown is clearly between the language and the usable criteria, as only 35% of headlines that pass the language criterion pass the usable criterion as well. Of those that do however, 68% pass the difficult final criterion.

Type	Language	Usable	Good
Real	1.0	0.89	0.89
Generated	0.87	0.35	0.68

Language	Usable	Good
1.0	0.89	0.79
0.87	0.31	0.21

Table 5: Summary for generated versus real headlines majority vote responses. The first table shows metrics for headlines that have passed the preceding criteria, while the second table shows the total for all headlines.

7 Discussions

Although free text generation is not the focus of this paper, the generative capabilities of the Finnish GPT are still noteworthy and relevant for the headline generation task. Evaluating the generative performance of a language model in-depth is a very time-consuming task, and we will outline the major findings we have with this particular model here. These findings mostly come from manually giving the model different prompts and using different parameterizations of top-p and top-k sampling to generate continuations.

From the logging of validation top-k next tokens and their assigned probabilities during training, it is clear that the output probability distribution for the next token becomes sharper as the training run progresses. The shape of the output distribution has a significant impact on sampling based decoding output, as sharp distributions produce less varied output. This makes generating a snippet during validation by using a fixed set of parameters for the sampling algorithm a poor way of gauging the progression of the true generative capability of a language model. Note that the temperature parameter directly affects the sharpness of the output distribution as well. For both top-k and top-p sampling,

we found that a range of 0.6-1.0 was the usable range for temperature. Values of over 1 result in very random and nonsensical text, while values of less than 0.6 became very repetitive.

Repetition is known as the most prevalent pathology in text generation using deep neural language models (Fu et al., 2021). This pathology occurs the worst the greedier the decoding algorithm. Greedy decoding and vanilla beam-search decoding which try to find the approximate MLE generation suffer from this the most. Top-k and top-p sampling partially combat this, by using the random nature of the sampling to break repetition loops. The true reason for the repetitive behavior of current language modeling solutions is not understood.

The first form of repetition is in the form of repeating entire or partial sentences one, several or even infinite times, sometimes with a slight variation. This makes for text that does not resemble human text, and is not desirable.

The second form of repetition is the repetition of names, places and objects in a way that does not semantically make sense. An engineered example: "*Sauli Niinistö tapasi keskiviikkona Tasavallan Presidentti Sauli Niinistön*" (Sauli Niinistö met the President of the Republic Sauli Niinistö on Wednesday). This sentence does not make sense, as a person cannot meet himself. In this case, it seems that the locally highly correlated continuation to "Tasavallan Presidentti" (President of the Republic), which is "Tasavallan Presidentti Sauli Niinistö" (the President of the Republic Sauli Niinistö) in the training data, overrides the fact that he should never be the prediction in this context conditioned on him being already mentioned in the sentence.

The opposite of repetition can occur. It can occur with greedier decoding as well but is more pronounced with sampling based decoding. Again, we class these into two main categories.

The first category is the direct opposite of the repetition of names and places. This is when a text mentions the name of a person, and the generated output suddenly swaps out the name for another name and continues the text with the new name. The severity of this varies depending on prompt length and context. If the context is very U.S. Presidential heavy and the name supplied is *Donald Bump*, it will likely be "corrected" to *Donald Trump* due to the sheer volume of support for the latter in the corpus.

Interestingly, the second form of correction may actually have some use. This is when a sentence is repeated, but with more probable grammar. As an example, there may be a grammatical error in a quote, the model can then accidentally correct the grammatical error when repeating the quote.

8 Conclusions

The task was to create and evaluate a headline generation algorithm in the context of helping editors in the newsroom in the creative process. This is what was done in this paper. A neural language model was pre-trained on Finnish text, and fine-tuned to generate headlines. A decoding algorithm for diverse output was implemented. The resulting generated headlines were evaluated by domain experts to gauge the feasibility of this model in actual use. This sort of evaluation is the first we've seen when it comes to evaluating a headline generation algorithm.

The final conclusions are that while most of the time the generated headlines are very close to being usable, this particular implementation is far from ready in any sort of automated system. This comes as no surprise, as even with near perfect usability performance it would still not be used without a human in the loop. The algorithm in this work has potential and an expressed interest as a creative aid for the headlining process.

The most common errors especially for the language and usable criteria are clear and have potential solutions. Some of the errors can be tackled by pre- and post-processing such as the unsightly special character code printouts. The repetition errors, which were the majority of language errors, can be reduced with the repetition penalty. We hypothesize that several of the errors could be tackled with an adversarial and/or active reinforcement learning approach. The problem with generative pre-training seems to be that the model is only trained with what is correct, with everything else being equally incorrect. In reality when producing headlines, this is not the case.

The next steps would be the low-hanging fruit: tackling the error types specifically with parsing fixes and repetition penalty, as well as letting the fine-tuning process converge more. After that, trying more strongly correlated metrics as the decoding algorithm base score, and trying encoder-decoder type approaches as well as active reinforcement learning or adversarial approaches.

Acknowledgments

Special thanks to Pipsa Havula (IS), Esa Mäkinen (HS) and Simo Holopainen (IS) from Helsingin Sanomat and Ilta-Sanomat for making the effort to evaluate the headline worksheet and provide feedback. Thanks to Helsingin Sanomat and Ilta-Sanomat for the corpus of news articles which constituted the bulk of the data used. Additionally we wish to thank the Finnish Computing Competence Infrastructure (FCCI) for supporting this project with computational and data storage resources. This work was partially financed by the Society of Swedish Literature in Finland with funding from Enhancing Conversational AI with Computational Creativity, and by the Ella and Georg Ehrnrooth Foundation for Modelling Conversational Artificial Intelligence with Intent and Creativity. This research has received mobility funding from Nokia Foundation under grant number 20220193.

References

- Abien Fred Agarap. 2018. [Deep learning using rectified linear units \(relu\)](#). *CoRR*, abs/1803.08375.
- Khalid Alnajjar, Leo Leppänen, and Hannu Toivonen. 2019. [No time like the present: Methods for generating colourful and factual multilingual news headlines](#). In *Proceedings of the 10th International Conference on Computational Creativity*, pages 258–265, Portugal. Association for Computational Creativity. International Conference on Computational Creativity ; Conference date: 17-06-2019 Through 21-06-2019.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Alexey Bukhtiyarov and Ilya Gusev. 2020. [Advances of transformer-based models for news headline generation](#).
- John William Charnley, Alison Pease, and Simon Colton. 2012. On the notion of framing in computational creativity. In *ICCC*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.
- Daniel Dor. 2003. [On newspaper headlines as relevance optimizers](#). *Journal of Pragmatics*, 35(5):695–721.
- Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76:378–382.
- Zihao Fu, Wai Lam, Anthony Man-Cho So, and Bei Shi. 2021. A theoretical analysis of the repetition problem in text generation. In *AAAI*.
- Mika Härmäläinen and Khalid Alnajjar. 2019. [Generating modern poetry automatically in Finnish](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5999–6004, Hong Kong, China. Association for Computational Linguistics.
- Mika Härmäläinen and Khalid Alnajjar. 2021a. The great misalignment problem in human evaluation of nlp methods. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 69–74, United States. The Association for Computational Linguistics.
- Mika Härmäläinen and Khalid Alnajjar. 2021b. Human evaluation of creative nlg systems: An interdisciplinary survey on recent papers. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*.
- Mika Härmäläinen, Khalid Alnajjar, Niko Partanen, and Jack Rueter. 2021. Never guess what i heard... rumor detection in finnish news: a dataset and a baseline. In *Proceedings of the Fourth Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*, page 39–44, United States. The Association for Computational Linguistics.
- Mika Härmäläinen and Timo Honkela. 2019. [Cooperation as an asymmetric form of human-computer creativity. case: Peace machine](#). In *Proceedings of the First Workshop on NLP for Conversational AI*, pages 42–50, Florence, Italy. Association for Computational Linguistics.
- Mika Härmäläinen and Jack Rueter. 2018. Development of an open source natural language generation tool for finnish. In *Proceedings of the Fourth International Workshop on Computational Linguistics of Uralic Languages*, pages 51–58, United States. The Association for Computational Linguistics.
- Dan Hendrycks and Kevin Gimpel. 2016. [Bridging nonlinearities and stochastic regularizers with gaussian error linear units](#). *CoRR*, abs/1606.08415.

- Jenna Kanerva, Samuel Rönqvist, Riina Kekki, T. Salakoski, and Filip Ginter. 2019. Template-free data-to-text generation of finnish sports news. In *NODALIDA*.
- Virapat Kieuvongngam, Bowen Tan, and Yiming Niu. 2020. [Automatic text summarization of covid-19 medical research articles using bert and gpt-2](#).
- Jeffrey Kuiken, Anne Schuth, Martijn Spitters, and Maarten Marx. 2017. [Effective headlines of newspaper articles in a digital environment](#). *Digital Journalism*, 5(10):1300–1314.
- Leo Leppänen, Myriam Munezero, Mark Granroth-Wilding, and Hannu Toivonen. 2017. [Data-driven news generation for automated journalism](#). In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 188–197, Santiago de Compostela, Spain. Association for Computational Linguistics.
- Yang Liu and Mirella Lapata. 2019. [Text summarization with pretrained encoders](#).
- Ilya Loshchilov and Frank Hutter. 2017. [Fixing weight decay regularization in adam](#). *CoRR*, abs/1711.05101.
- Kazuki Matsumaru, Sho Takase, and Naoaki Okazaki. 2020. [Improving truthfulness of headline generation](#). *CoRR*, abs/2005.00882.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Alec Radford and Karthik Narasimhan. 2018. [Improving language understanding by generative pre-training](#). In *arxiv*.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. [Language models are unsupervised multitask learners](#). *OpenAI blog*.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Neural machine translation of rare words with subword units](#).
- Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. 2012. [Practical bayesian optimization of machine learning algorithms](#).
- Kaiqiang Song, Bingqing Wang, Zhe Feng, Ren Liu, and Fei Liu. 2020. [Controlling the amount of verbatim copying in abstractive summarization](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:8902–8909.
- Ottokar Tilk and Tanel Alumäe. 2017. [Low-resource neural headline generation](#). In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 20–26, Copenhagen, Denmark. Association for Computational Linguistics.
- Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). *ArXiv*, abs/1706.03762.
- Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. 2016. [Diverse beam search: Decoding diverse solutions from neural sequence models](#). *CoRR*, abs/1610.02424.
- Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. [Pegasus: Pre-training with extracted gap-sentences for abstractive summarization](#).

Generating Coherent and Informative Descriptions for Groups of Visual Objects and Categories: A Simple Decoding Approach

Nazia Attari

Research Institute for Cognition and Robotics
Bielefeld University, Germany
nattari@techkfak.uni-bielefeld.de

David Schlangen

Computational Linguistics
University of Potsdam, Germany

Martin Heckmann

Aalen University, Germany

Heiko Wersing

Honda Research Institute Europe, Germany

Sina Zarriß

Faculty for Linguistics and Literature Studies
Bielefeld University, Germany

Abstract

State-of-the-art image captioning models achieve very good performance in generating descriptions for instances of visual categories and reasoning about them, e.g. imposing distinctiveness of the description in the context of distractors. In this work, we propose an inference mechanism that extends an instance-level captioning model to generate coherent and informative descriptions for groups of visual objects from the same or different categories. We test our model in the domain of bird descriptions. We show that group-level descriptions generated by our method are (i) coherent, pulling together properties that are true for all or majority of its instances, and (ii) informative, as they allow an external BERT-based text classifier to identify the target category more accurately in comparison to single-instance captions and are preferred by human evaluators.

1 Introduction

State-of-the-art image captioning models excel at generating semantically accurate descriptions of single images (Anderson et al., 2018; Cornia et al., 2020) and can be enhanced with communicative-pragmatic reasoning procedures that impose distinctiveness of the description in the context of distractors at inference time (Vedantam et al., 2017; Cohn-Gordon et al., 2018; Zarriß and Schlangen, 2019). To date, however, discriminative image captioning has been restricted to informative *instance* descriptions and has not yet explored descriptions for *groups* (or sets) of objects – a classical problem in referring expression generation (REG) (Stone, 2000; Gardent, 2002; Horacek, 2004; Gatt, 2007; Krahmer and van Deemter, 2011). In this paper,

we investigate whether an instance-level captioning model can be extended to generate coherent and informative descriptions for groups of visual instances, by integrating communicative-pragmatic reasoning at inference time.

Generating a description for a group of visual entities require optimizing two objectives: (i) coherence, i.e., the description should pull together properties that are true for all or most of the groups’ instances and (ii) informativeness, i.e., it should mention those properties that are distinctive in a particular context (Gatt, 2007). Krahmer and van Deemter (2011) point out that the traditional Incremental Algorithm for symbolic REG directly applies to sets of entities, when they have certain properties in common. In this paper, we test whether this also holds for neural captioning models and propose a simple task, an inference scheme and experimental protocol for generating group-level descriptions. In particular, we extend the emitter-suppressor beam search by Vedantam et al. (2017) with an additional, simple coherence objective.

The ability to generate descriptions of groups is not only relevant for reference but also for explanation tasks, which become increasingly important in machine learning (Ribeiro et al., 2016; Lundberg and Lee, 2017). Here, systems commonly need to verbalize their knowledge about the shared properties of instances in a category, for instance, when learning to classify birds in images (Hendricks et al., 2016). However, an instance-based explanation might produce a rather idiosyncratic description of an image rather than a more representative description of the categories (that is true for majority of instances in a set). This becomes cru-

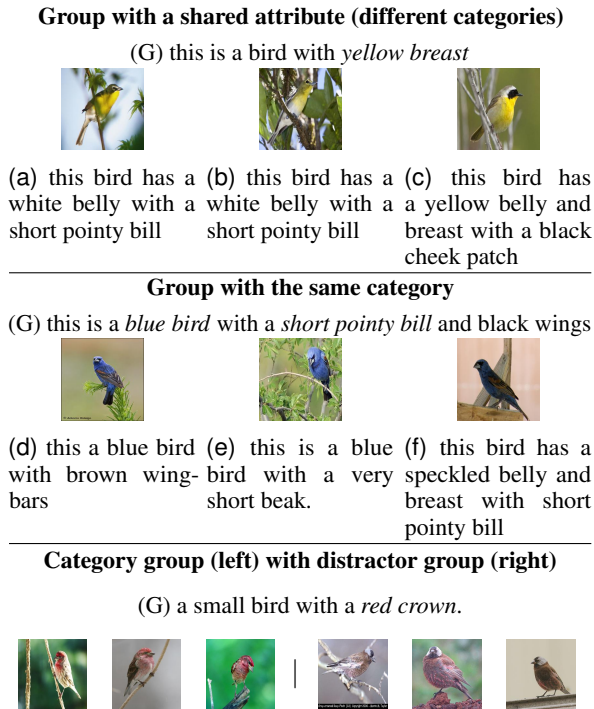


Figure 1: Examples of generated group (G) and instance (a-f) descriptions for types of bird groups.

cial in scenarios where a system needs to describe to a user the difference between two categories of birds. Here, it does not suffice to characterize only a single-instance, but, ideally, the system should preferably have a linguistic component explaining its knowledge about the category (Figure 1: middle section). Thus, for our study, we use the Caltech UCSD birds data (Wah et al., 2011) that provides fine-grained categories for bird images and descriptions of instances (Reed et al., 2016), and has been leveraged for instance-level explanation generation by Hendricks et al. (2016). In the context of captioning images of birds, we show that our approach to group-level decoding can be used for different types of groups and corresponding descriptions: (i) **objects with a shared attribute** but from different categories (i.e. bird species) and (ii) **objects of the same category**, sharing multiple visual properties, as shown in Figure 1. We assess the quality, coherence and informativeness of these group descriptions in human and automatic evaluation, including a set up for category prediction based on generated descriptions.

2 Related work

Research on language generation from visual inputs often builds upon generic image captioning models that are trained to produce “neutral” de-

scriptions for images depicting instances of objects or scenes (Vinyals et al., 2015; Xu et al., 2015; You et al., 2016; Rennie et al., 2017; Anderson et al., 2018; Hossain et al., 2019; Cornia et al., 2020; Zhou et al., 2020; Luo et al., 2021). One line of work has extended captioning towards more complex visual inputs, e.g., sequences of images depicting events or stories (Yu et al., 2017; Mun et al., 2019; Gao et al., 2020). Other work has looked at enhancing captioning models towards generating more informative outputs that fulfill specific communicative goals, by leveraging contextual and contrasting information along with the target image at inference time (Vedantam et al., 2017; Cohn-Gordon et al., 2018; Zarri  and Schlangen, 2019; Nie et al., 2020). Our work connects these two lines by extending Vedantam et al. (2017)’s discriminative instance-level decoding scheme for groups of image instances.

Our task and set-up is similar to Li et al. (2020)’s work on context-aware group captioning, where the goal is to build a model that captions a group of images with a matching scene graph (e.g. *women in chair*) in the context of a more general reference set of images (e.g. *women*). Their approach rests on a supervised model that is trained on a dataset of group captions (compiled from instance captions) and that performs group-wise visual feature aggregation with self-attention and contrastive visual feature construction. While Li et al. (2020) investigate rather short group descriptions for common objects (e.g., *women with hat*) with an average caption length of around 3, we test our approach on bird descriptions which involves a careful selection of properties for informative and coherent descriptions that have an average length greater than 10. Moreover, our work aims at describing groups by reasoning at the word level about which words can be used to refer to the group’s instances, without retraining the underlying captioning model.

In comparison to earlier work on REG for sets, though, our approach targets rather simple descriptions of groups that essentially mention the properties that hold for the members of the set. Thus, we do not address more complex linguistic phenomena such as plurals, coordination, disjunction, or quantification. Gatt (2007), for instance, investigates conceptual coherence for the generation of sets whose entities cannot be referred to by the same head noun, triggering a competition between coordinations like *the chef and the engineer* and

the Italian and the Frenchman. As our approach assumes that group descriptions can be decoded from an instance-level captioning model, it will not be able to generate linguistic structures that do not appear in the training data of that captioning model. For instance, phenomena like coordination do not appear in our descriptions which typically enumerate properties of a single bird, named *bird*, see Figure 1. This is the case for our model as it uses bird description data where all captions refer to a single bird, which is named *bird*, see Figure 1.

3 Approach

This section defines the task and decoding procedures for coherent and informative group-level captioning.

3.1 Task Description

We assume to be given a dataset that pairs image instances i with verbal descriptions s and some category information c . We also assume that a captioning model of some sort, which we refer to as speaker $S(I)$, is trained on this data and predicts the probability of sequences of words given a single image $p(s|i)$.

We frame the task of generating group descriptions as a decoding or inference task, where the input to the model is a target group of n instances, $G_t = \{i_1, i_2, \dots, i_n\}$ and the goal is to predict $p(s|G_t)$ based on the speaker $S(I)$, without any further training or fine-tuning of the instance-level captioning.

This basic group description task can be extended towards a discriminative description task where the model receives an additional context, i.e. a distractor group of $G_d = \{i_1, i_2, \dots, i_m\}$. In discriminative group description decoding, the goal is to predict a pragmatically informative sequence of words s such that a listener can distinguish the target from the distractor group or, more formally, such that $p(G_t|s) > p(G_d|s)$.

3.2 Coherent Group Decoding

The objective of the basic group-level speaker $S(G_t)$ is to maximize the probability of the output sequence given all images in the target group:

$$S(G_t) = \operatorname{argmax}_s \frac{1}{n} \sum_{l=1}^n \log p(s|i_l) \quad (1)$$

As the space over possible output sequences s cannot be searched exhaustively, we approximate

this objective via beam search: at every time step, we (i) input all instances of the group to speaker $S(I)$ in parallel, (ii) compute the mean of log-probabilities over the entire vocabulary of all instances of the group and (iii) put the top- k words on the beam, as input to the next time-step. The stepwise averaging over log word probabilities directly implements the idea of coherence, i.e. the model should verbalize the common properties that are likely for all instances in the group.

3.3 Discriminative Group Decoding

We expect that $S(G_t)$ produces descriptions that summarize common properties of a group, but that it may not always select particularly informative properties that accurately discriminate a group in context. Thus, we define the discriminative group speaker $S(G_d)$ for instances in the distractor group, with the following objective:

$$S(G_d) = \operatorname{argmax}_s \frac{1}{m} \sum_{k=1}^m \log p(s|i_k) \quad (2)$$

We use $S(G_d)$ to induce discriminativeness of the output by combining it with $S(G_t)$ and reconstructing the emitter-suppressor beam objective by Vedantam et al. (2017) for groups:

$$S(G_{t,d}) = S(G_t) - (1 - \lambda) \cdot S(G_d) \quad (3)$$

$S(G_{t,d})$ is the group speaker that maintains a trade-off between coherence and informativeness of the generated sequences, and can be pushed towards higher discriminativeness with appropriate values of the λ parameter.

The speakers in Equation 3 can be further factorized, incorporating word probabilities for the sequence as $\prod_{\tau=1}^T p(s_\tau|s_{1:\tau-1}, I)$, where T is the length of the sentence. Hence, we obtain the following objective for our inference mechanism:

$$S(G_{t,d}) = \operatorname{argmax}_s \sum_{\tau=1}^T \frac{1}{n} \sum_{l=1}^n \log p(s_\tau|s_{1:\tau-1}, i_l) - (1 - \lambda) \cdot \left(\frac{1}{m} \sum_{k=1}^m \log p(s_\tau|s_{1:\tau-1}, i_k) \right) \quad (4)$$

Again, we approximate this objective via beam search. At every time-step, we subtract the average

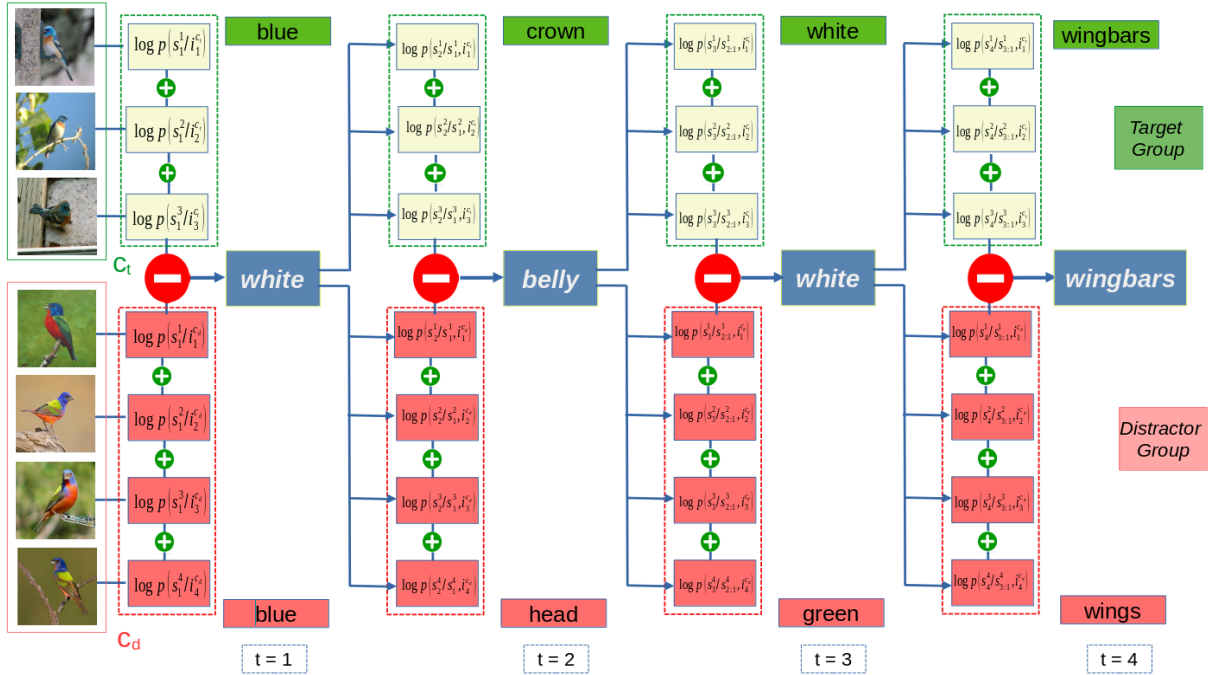


Figure 2: Illustration of an example phrase generated by discriminative group-level decoding with beam size 1 (*white belly white wingbars* in blue boxes). The decoding scheme favours coherent and discriminative properties over less discriminative ones predicted by (target only) group-level decoding (*blue crown* in top green boxes)

log probability of a word for the target instances by its log probability for the distractor instances. Words that have high probability for the in-group images and low probability for the out-group images will be more likely to be put on the beam than words that are equally probable for both, or even more probable for the out-group.

We demonstrate the mechanics of our decoding procedure in Figure 2, with a beam size of 1 for simplicity. It shows how the speaker S_t , at inference time, combines probability scores of the respective groups and produces the best possible output words. In our experiments, we used a beam size of 10.

4 Experimental Set-up

4.1 Data

We base our work on the CUB-200-2011 dataset (Wah et al., 2011), originally designed for subordinate category categorization, detection and part localization. It contains 11788 images of 200 North American bird species and every species has approximately 60 image instances. Each image instance is characterized by 28 symbolic attributes using an online tool for bird identification¹ curated by bird experts, further leading to an extensive set of human-annotated 312 binary attribute-value

¹www.whatbird.com

pairs (e.g. *beak-shape:hooked*, *belly-color:white*, *tail-pattern:spotted*). For our first experiment, we used this symbolic information to form groups of image instances from different bird categories.

We also have access to (five) textual descriptions for each image instance collected by Reed et al. (2016); the annotators were asked to mention the physical bird attributes (wing color, beak shape, body color and so on) visible in the image without any reference to the bird species and using basic vocabulary unlike sophisticated expert-level vocabulary. We note, however, that in some cases, the annotators also mentioned non-discriminating properties, for instance, where the bird is looking at, it's flying or sitting.

4.2 Sampling Groups

The target groups (G_t) in our experiments can be of two kinds: (i) groups of instances from different bird categories with a shared attribute, (ii) groups of instances from the same bird category. For the latter, we induce additional context from a similar distractor group (G_d), composed of instances from a distractor bird category. We use distractor categories that belong to the same bird family as the target, for instance, *Black-footed-Albatross* and *Laysan-Albatross* from the bird family *Albatross*, similar to Vedantam et al. (2017), to test whether

we can generate informative descriptions in challenging contexts.

For training the instance-level speaker (S), we used the split as provided by Hendricks et al. (2016) (train:4000, val:1994, test:5794). For our shared attribute grouping, we sample a target group of size 3 for every instance in each split, such that we obtain 1994 and 5794 groups for val and test. For category-level grouping, we sample target groups of size 3 and distractor groups of size 4 for each instance. There are 7 bird species that do not have distractors from the same family and we ignore these here. For discriminative group decoding, we obtain 3358, 1646, and 4833 groups for train, val and test respectively.

4.3 Model

We first train an image classifier by finetuning a pretrained resnet-101 architecture to predict bird categories from bird images. The training parameters were set to: batch size 16, (RGB) image size as 448, learning rate 0.001 for a total of 50 epochs with a decay factor of 0.1 after every 20 epochs. We use this image classifier as visual encoder for our speaker $S(I)$, the image captioning model.

We trained two versions of our speaker $S(I)$, (i) a basic recurrent **LSTM** model architecture from Xu et al. (2015) and (ii) a basic **Transformer** by (Vaswani et al., 2017). Generally, transformers are currently the more popular model due to their parallel processing and multi-head attention architecture (Devlin et al., 2019; Lan et al., 2020), but they may also be more data-intensive. We wanted to see how both architectures (LSTM and Transformer) perform given our dataset is quite small. Both models use the visual encoder described above. Both our captioning models, the LSTM and Transformer, achieve similar CIDEr-D validation scores of 49.4 and 49.5 respectively, similar to existing captioning models for the birds data (Vedantam et al., 2017).²

5 Experiment 1: Shared Attributes

In this experiment, we investigate whether our group decoding mechanism can be used to systematically include a shared visual attribute in a description for group of instances (which may belong to a different bird categories).

²Code and models can be found [here](#)

5.1 Attributes

We sample groups with shared attributes based on the symbolic attribute annotations in the birds data. We use the attribute-value pair as a reference pattern that needs to be included in an accurate, coherent group description (e.g. for *belly-color:white* we look for *white belly*). It is important to note that the symbolic attribute annotations are significantly more detailed and elaborate in terms of their vocabulary (Section 4.1) than the captions which were crowd-sourced with non-experts. This results in a mismatch between aspects of birds that are annotated and properties that are verbalized in the captions and that we can expect the captioning model to be able to pick up. To tackle this, we restricted our group sampling to attributes that can be detected in captions by simple pattern search. We ranked the symbolic attributes present in the captions by frequency and selected randomly four more frequent and two less frequent attributes for our experiment. For simplicity, we used only one shared attribute per group at a time and no distractors, as we expect to obtain rather noisy distractor sets due to above mentioned issues with the attribute annotations.

5.2 Results

We assess the accuracy of decoding for groups with a shared attribute, i.e. whether the output description contains the shared attribute as identified by pattern search. Table 1 shows that for 4 out of 6 selected attribute-value pairs, group captions are clearly more likely to mention the selected common property than the instance captions, with increase in accuracy of up to 17% for *bill length*. We also note that the instance-level captions generated by the LSTM and Transformer show differences in their attribute patterns, despite their overall similar performance. We will discuss differences between the two models further below. Figure 3 shows a qualitative example where the group descriptions mention the shared property (*blue wing*) in contrast to all the instance descriptions.

For the less frequent attribute *bill shape* in the ground-truth instance captions, and not so distinctive attribute *eye color* (as most of the times it's value is black), the accuracy is low for both instance and group-level decoding and the instance-level decoding outperforms the group-level decoding for the *bill shape* attribute. This suggests that achieving coherence in group descriptions in decoding is contingent on shared properties occurring with



(a) this bird has a white belly and a breast with a blue belly and a blue crown
 (b) this is a small bird with a white head
 (c) this bird has a blue crown and a blue long bill

$S(G_t)$ -LSTM: this is a bird with **blue wings**.

$S(G_{t,d})$ -Transformer: this is a bird with **blue wing**.

Figure 3: Generated instance and group caption for a shared-attribute group.

a certain frequency in the instance caption data, or, vice versa, that the group decoding may not push the captioning models towards selecting rare attribute words and fine-grained visual details.

Shared Attributes	Frequency (total)	Mentions of shared attribute(%)			
		LSTM		Transformer	
		group	instance	group	instance
<i>breast color</i>	10158	50	35.40	25.95	12.30
<i>crown color</i>	9693	31.57	20.57	38.61	19.59
<i>belly color</i>	9379	47.67	34.85	25.00	14.62
<i>eye color</i>	8666	14.86	10.06	19.08	16.22
<i>bill length</i>	7372	61.63	44.61	56.08	41.63
<i>bill shape</i>	6882	7.61	11.54	15.76	23.86

Table 1: Accuracy of generated group captions and instance captions in terms of mentioning a shared attribute. Frequency shows occurrence of a shared attribute in original captions.

6 Experiment 2: Category-level Grouping

In the second experiment, we test whether our decoding mechanism generates coherent and informative descriptions of groups that correspond to categories, i.e. instances are sampled based on category-level annotation in the birds dataset and, optionally, paired with distractor groups/categories.

6.1 Evaluation

Evaluation is challenging as we do not have vision-oriented ground-truth category descriptions. Expert-level category definitions from, e.g., bird dictionaries would not help to objectively assess our group descriptions as they use a more sophisticated vocabulary and commonly mention non-visual properties that cannot be learned by a captioning model. Therefore, we combine automatic evaluation based on automatically selected, prototypical reference descriptions, automatic category inference and human evaluation on the most promising models.

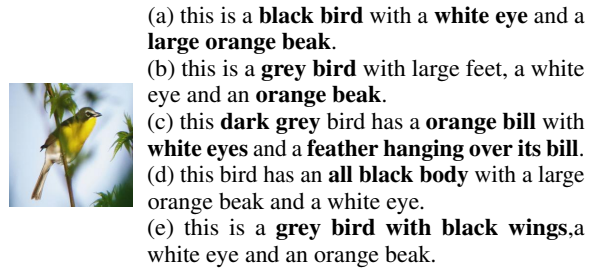


Figure 4: Five most similar instance descriptions for a bird category based on cosine similarity to centroid.

General quality of group captions We want to ensure that we do not lose lexical richness by moving from instance to group descriptions, as these problems have commonly been observed in neural NLG models. We computed average sentence length and Dist-k (Ippolito et al., 2019) (distinct unigrams and bigrams) to measure lexical diversity and repetitiveness in generated captions.

Prototypical reference captions We compile the reference set for a category by taking the top-5 prototypical descriptions for each bird category. We select these descriptions using kmodes clustering (de Vos, 2015) on pre-trained BERT sentence embeddings. We compute the centroid of the bird description embeddings and take the five most similar instance descriptions (according to cosine-similarity) as a stand-in for general, prototypical descriptions for the target category. Figure 4 shows an example of the top-5 descriptions determined by the clustering algorithm. It shows that they cover distinctive representative parts of a bird category, thereby getting rid of the erroneous (non-discriminating) descriptions.

Overlap with target and distractor references

We use two standard overlap metrics, BLEU-4 (Papineni et al., 2002) and CIDEr (Vedantam et al., 2015), to assess the similarity of generated group descriptions with reference sentence groups. We re-purpose these for: (i) for **target-target similarity**, i.e. measuring the overlap of generated group descriptions to a set of references for the target group, (ii) for **target-distractor similarity**, i.e. measuring the overlap of generated group descriptions to a set of references for the *distractor group*. We expect the target-target similarity to go up for group captions and the target-distractor similarity to go down for group captions that are informative.

Category-level Inference In order to verify that the generated group descriptions indeed pull to-

gether properties relevant for the target category and make it distinct from the other distractor category, we learn an external text classifier based on BERT (Devlin et al., 2019). As we do not have ground-truth category descriptions, we use the generated group captions from our different group decoding methods and for a fair comparison, generated instance captions from the speaker $S(I)$ for training. The performance of these text classifiers give us some indication as to whether using group of instances during decoding leads to descriptions that make it easier to identify the target category, as compared to descriptions for single instances, in the absence of concrete visual instances. This resembles a setting where a speaker explains to a listener the properties that it has learned to detect for a given category. As for the training parameters of the text classifier, we set the batch size to 64 and learning rate to 0.00002 for a total of 60 epochs.

Human Evaluation We performed human evaluation on the most promising LSTM and Transformer speaker models using the Amazon Mechanical Turk (AMT) crowdsourcing platform, in order to analyze whether group descriptions were preferred over instance descriptions for describing a group of image instances. We showed the participants all images from the target group and two competing descriptions: (A) the generated discriminative group description and (B) the generated instance description (from a random instance in the group). We asked them to carefully observe the images and select the description(s) that best describe all or most of the images in the group in a forced-choice task with 3 options, (A), (B) or (C) both. We included the third choice as we observed that the instance descriptions in the birds data can be very similar to the prototypical description of the target category and we wanted to avoid random choices by participants for these cases. We randomly selected 2 groups out of 60 bird categories, having a total of 120 group and instances descriptions. More details on the set-up are provided in Appendix B.

6.2 Results

Figure 5 shows generated descriptions produced for instances and groups using category-level decoding with both LSTM and Transformer based speakers.

Quality and Overlap Metrics Table 2 reports the automatic overlap metrics for target-target similarity and for target-distractor similarity for differ-

ent models and decoders. These results indicate that there is a general positive tendency towards higher target-target similarity and lower target-distractor similarity when using group-level instead of instance-level decoding. Another general tendency is that the difference between the instance-level decoding and the coherence-only group decoding (with distractors) is rather subtle and that the real gain comes from combining the coherence and discrimination objective, i.e. CIDEr scores for target-target similarity increase from 68 to 81 and 79 to 88 for the LSTM and Transformer when used with $S(G_{t,d})$ instead of $S(G_t)$ (the λ parameter needs to be set differently with the two captioning models). CIDEr also predicts a rather sharp decrease of target-distractor similarity for the transformer-based decoding (47 to 36), but less of a decrease for the LSTM-based discriminative group decoding. This suggests that captions decoded on the group-level are more likely to mention properties that are both more coherent and informative for the target category. CIDEr scores show a big positive effect for using discriminative group-level decoding with the LSTM and the Transformer on target-target similarity, whereas the BLEU-4 score indicates a smaller increase. Furthermore, CIDEr indicates a strong difference for instance-level decoding between LSTM und Transformer, whereas BLEU-4 favours instance-level captions generated by the LSTM (in terms of their similarity to the group reference). For this reason, we complement this type of evaluation with further assessments below. Finally, we find that the average sentence length and the dist-k scores are high for the instance and for category descriptions, as shown in Table 2. This shows our group-based decoding does not lead to negative effects regarding length or repetitiveness which have been observed for other decoding methods in neural NLG (Ippolito et al., 2019; Zarri  and Schlangen, 2018).

Category-level Inference Table 3 shows accuracy results for text classifiers trained to identify the bird category based on generated captions. We find that coherent group decoding improves the prediction of target categories and discriminative decoding enhances the classifier further. Moreover, this evaluation indicates the superior performance of the Transformer over the LSTM speaker, in line with the CIDEr evaluation in Table 2. This suggests that the power of the underlying captioning model, which may not become apparent in instance-

Model	Decoding	λ	Target-target sim. (\uparrow)		Target-distractor sim. (\downarrow)		Diversity		
			BLEU-4	CIDEr	BLEU-4	CIDEr	Dist-1	Dist-2	avg. len
LSTM	$S(I)$	-	42.41	68.89	36.56	44.97	0.88	0.98	12.96
	$S(G_t)$	-	42.54	68.11	36.70	44.59	0.89	0.98	13.01
	$S(G_{t,d})$	0.3	45.11	81.32	34.04	40.86	0.86	0.97	12.91
	$S(G_{t,d})$	0.5	44.55	78.21	36.10	43.79	0.88	0.98	12.97
Transf	$S(I)$	-	40.68	77.44	32.89	47.02	0.89	0.98	13.29
	$S(G_t)$	-	41.16	79.45	32.45	44.46	0.90	0.99	13.27
	$S(G_{t,d})$	0.3	42.62	83.79	28.58	36.96	0.84	0.96	13.36
	$S(G_{t,d})$	0.5	43.69	88.87	31.27	41.54	0.88	0.98	13.31

Table 2: Evaluation of category-level group captions for overlap with prototypical target and distractor references. Decoding: $S(I)$ instance-level, $S(G_t)$ coherent group decoding, $S(G_{t,d})$ discriminative group decoding.

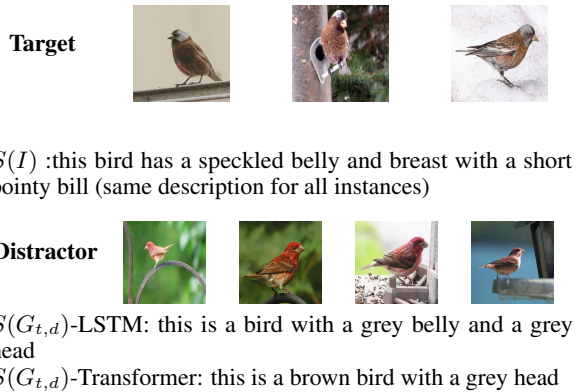


Figure 5: Generated group caption for category.

level use, is important for high-quality group-level decoding. In future work, we plan to further analyze the interaction of the underlying captioning architecture with the decoding mechanism.

Model	Decoding	λ	Accuracy
LSTM	$S(I)$	-	18.22
	$S(G_t)$	-	19.70
	$S(G_{t,d})$	0.3	33.14
	$S(G_{t,d})$	0.5	25.59
Transformer	$S(I)$	-	23.60
	$S(G_t)$	-	29.48
	$S(G_{t,d})$	0.3	42.72
	$S(G_{t,d})$	0.5	36.90

Table 3: Text classification performance for category identification. Discriminative group decoding $S(G_{t,d})$ leads to best performance for LSTM and Transformer.

Human Evaluation Table 4 shows that participants prefer group over instances descriptions for the LSTM and Transformer model for 59% of the items. Again, we see that the instance-level Transformer outperforms the LSTM, i.e. there are fewer

Transformer captions where participants rate the instance and group-level caption equally. Generally, this clearly supports our hypothesis that group-level decoding can pull together multiple distinctive properties common to a group or category.

Model	Selected by participants (%)		
	$S(I)$	$S(G_{t,d})$	<i>Both</i>
LSTM	9.17	59.17	31.67
Transformer	17.5	59.17	23.33

Table 4: Human evaluation with portion of items where participants selected generated instance-level, group-level or both captions as appropriate for a group.

6.3 Limitations

As our approach to decoding group-level descriptions is conceptually simple, it is not surprising that it has certain limitations in terms of the linguistic phenomena it is able to account for. Figure 6 shows examples for systematic limitations (and directions for future work): (i) **describing discriminative details**: for some bird families, the effect of group decoding is not significant and fixating fine-grained details is not yet possible, see the *sparrow* example in Figure 6’s first row. (ii) **completeness**: group descriptions do not always mention all the properties that might be used to define a category because the distractor group has similar properties, in Figure 6 third row, *black on its wings* was ignored due to the distractor group. (iii) **disjunctive properties within a category**: different physical appearance of male and female instances of a bird species leads to incoherent captions as in Figure 6 second row.

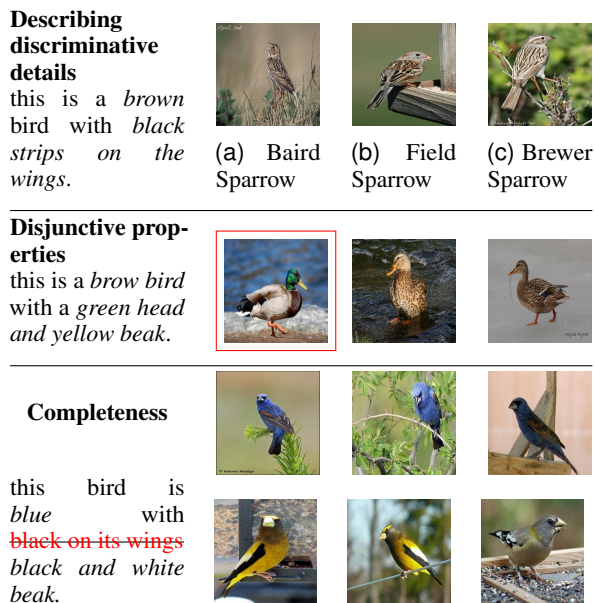


Figure 6: Examples for errors and limitations.

7 Conclusion

In this paper, we have proposed a task, a set-up and a decoding procedure for generating group-level descriptions with an instance-level captioning model. Despite our decoding approach being arguably simple, the results are encouraging and point into some interesting directions for future work. The classical problem of REG could be re-visited on a larger scale for sets of “real-world” objects or one could explore the use of group decoding in explanation scenarios where additional category label information or predicted attention maps could be integrated to provide post-hoc justifications. Finally, enhancing the decoding mechanism with deeper logical reasoning capabilities (e.g. on disjunctions) seems to be a promising direction.

References

- Peter Anderson, X. He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
- Reuben Cohn-Gordon, Noah Goodman, and Christopher Potts. 2018. Pragmatically informative image captioning with character-level inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 439–443.
- Marcella Cornia, Matteo Stefanini, L. Baraldi, and R. Cucchiara. 2020. Meshed-memory transformer for image captioning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Nelis J. de Vos. 2015. kmodes categorical clustering library. <https://github.com/nicodv/kmodes>.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics.
- Lianli Gao, Xiangpeng Li, Jingkuan Song, and Heng Tao Shen. 2020. Hierarchical lstms with adaptive attention for visual captioning. *IEEE Trans. Pattern Anal. Mach. Intell.*
- Claire Gardent. 2002. Generating minimal definite descriptions. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 96–103.
- Albert Gatt. 2007. *Generating coherent references to multiple entities*. Ph.D. thesis, Citeseer.
- Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. 2016. Generating Visual Explanations. In *ECCV*.
- Helmut Horacek. 2004. On referring to sets of objects naturally. In *International Conference on Natural Language Generation*, pages 70–79. Springer.
- MD. Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. 2019. A comprehensive survey of deep learning for image captioning. *ACM Comput. Surv.*
- Daphne Ippolito, Reno Kriz, João Sedoc, Maria Kustikova, and Chris Callison-Burch. 2019. Comparison of diverse decoding methods from conditional language models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.
- Emiel Kraemer and Kees van Deemter. 2011. Computational generation of referring expressions: A survey. *Computational Linguistics*, 38.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. *ArXiv*, abs/1909.11942.
- Zhuowan Li, Quan Hung Tran, Long Mai, Zhe Lin, and A. Yuille. 2020. Context-aware group captioning via self-attention and contrastive features. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3437–3447.
- Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

- Yunpeng Luo, Jiayi Ji, Xiaoshuai Sun, Liujuan Cao, Yongjian Wu, Feiyue Huang, Chia-Wen Lin, and Rongrong Ji. 2021. [Dual-level collaborative transformer for image captioning](#). *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Jonghwan Mun, L. Yang, Zhou Ren, N. Xu, and Bohyung Han. 2019. Streamlined dense video captioning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Allen Nie, Reuben Cohn-Gordon, and Christopher Potts. 2020. [Pragmatic issue-sensitive image captioning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1924–1938, Online. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.
- Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. 2016. Learning Deep Representations of Fine-Grained Visual Descriptions. In *CVPR*. IEEE Computer Society.
- Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. [Self-critical sequence training for image captioning](#). In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.
- Matthew Stone. 2000. On identifying sets. In *INLG'2000 Proceedings of the First International Conference on Natural Language Generation*, pages 116–123.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*.
- Ramakrishna Vedantam, Samy Bengio, Kevin Murphy, Devi Parikh, and Gal Chechik. 2017. Context-aware captions from context-agnostic supervision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *CVPR*. IEEE Computer Society.
- Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. [Show and tell: A neural image caption generator](#). In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. 2011. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology.
- Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2048–2057.
- Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. [Image captioning with semantic attention](#). In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Licheng Yu, Mohit Bansal, and Tamara L. Berg. 2017. Hierarchically-attentive rnn for album summarization and storytelling. In *EMNLP*.
- Sina Zarrieß and David Schlangen. 2018. [Decoding strategies for neural referring expression generation](#). In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 503–512, Tilburg University, The Netherlands. Association for Computational Linguistics.
- Sina Zarrieß and David Schlangen. 2019. [Know what you don't know: Modeling a pragmatic speaker that refers to objects of unknown categories](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 654–659, Florence, Italy. Association for Computational Linguistics.
- Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. 2020. [Unified vision-language pre-training for image captioning and vqa](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34.

A Automatic Evaluation II

We used another reference set for automatic evaluation of category-level group descriptions which is union of ground-truth descriptions of all instances in a group. This amounts to 15 reference captions for the target group of size 3 in our case and 20 reference captions for the distractor group as distractor group size is 4. We observe that group of ground-truth instance descriptions are still to some extent composed of idiosyncratic properties of instances. This can be seen from small amount of increase in CIDEr scores from instance to discriminative group descriptions in Table 5 compared to that of in Table 2 using prototypical references.

B Crowdsourcing Details

In this section, we provide additional information on how we conducted the human evaluation using AMT crowdsourcing platform. We recruited

Model	Decoding	λ	Target-target sim. (\uparrow)		Target-distractor sim. (\downarrow)	
			BLEU-4	CIDEr	BLEU-4	CIDEr
LSTM	$S(I)$	-	62.85	42.73	61.27	28.08
	$S(G_t)$	-	64.01	44.38	63.15	29.61
	$S(G_{t,d})$	0.3	63.67	46.24	55.81	23.49
	$S(G_{t,d})$	0.5	64.57	46.84	60.18	26.91
Transf	$S(I)$	-	58.30	43.17	54.57	25.95
	$S(G_t)$	-	60.09	45.19	56.41	27.31
	$S(G_{t,d})$	0.3	57.30	43.58	46.03	19.58
	$S(G_{t,d})$	0.5	60.57	47.50	52.30	23.78

Table 5: Evaluation of category-level group captions for overlap with union of ground-truth instance descriptions from target and distractor groups.

participants who are native english speakers (e.g., from United Kingdom, United States) as our task requires English proficiency. We paid the participants 0.15\$ for successfully completing the task based on a fair hourly wage. Figure 7 shows an example of how our task was presented to the participants. The participants were aware that the task is purely for research purposes and contains no form of controversial data.

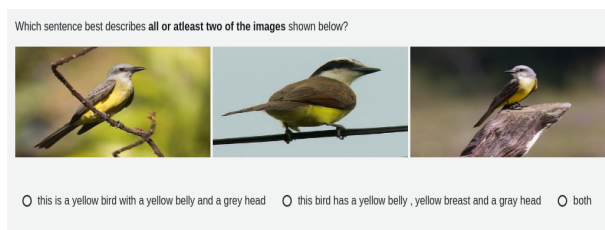


Figure 7: An example of the task seen by the participants on AMT platform.

Dealing with hallucination and omission in neural Natural Language Generation: A use case on meteorology

Javier González-Corbelle, Jose M. Alonso-Moral, A. Bugarín-Diz
Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS),
Universidade de Santiago de Compostela, Spain

{j.gonzalez.corbelle, josemaria.alonso.moral, alberto.bugarin.diz}@usc.es

J. Taboada

MeteoGalicia

Xunta de Galicia

Santiago de Compostela, Spain

coordinador-prediccion.meteogalicia@xunta.gal

Abstract

Hallucinations and omissions need to be carefully handled when using neural models for performing Natural Language Generation tasks. In the particular case of data to text applications, neural models are usually trained on large-scale datasets and sometimes generate text including divergences with respect to the input data. In this paper, we show the impact of the lack of domain knowledge in the generation of texts containing input-output divergences through a use case on meteorology. To analyze these phenomena we adapt a Transformer-based model to our specific domain, i.e., meteorology, and train it with a new dataset and corpus curated by meteorologists. Then, we perform a divergences' detection step with a simple detector in order to identify the clearest divergences, especially those involving hallucinations. Finally, these hallucinations are analyzed by an expert in the meteorology domain, with the aim of classifying them by severity, taking into account the domain knowledge.

1 Introduction

Since the emergence of Natural Language Generation (NLG), this subfield of Natural Language Processing (NLP) has not stopped evolving. However, the fastest evolution has occurred in the last years, due to the advances made in Deep Learning models. With the arrival of the attention mechanism and the Transformer-based models (e.g., BERT [Devlin et al., 2019], GPT-2 [Radford et al., 2019], or GPT-3 [Brown et al., 2020]), the way in which NLG tasks such as text summarization, question answering, or data to text (D2T) are approached has changed drastically. Before the appearance of these end-to-end neural models, the generation of

NLG models had at least two main tasks to accomplish (content selection and surface realization) and sometimes even more subtasks (e.g., lexicalization or aggregation) (Reiter and Dale, 1997). Now, with end-to-end models, the whole generation process is made in a single step. Furthermore, neural models allow us to obtain natural, diverse, and fluent texts. Of course, these models also have their drawbacks, such as the necessity of a large corpus or enough computational resources to train the model for a given task.

In addition, in the context of D2T, texts generated by neural models are sometimes affected by divergences with the input data (Dušek et al., 2019). On the one hand, neural models may generate texts that are incoherent or unrelated with the input of a D2T system, i.e., hallucinations. On the other hand, generated texts may not mention some (relevant) information from the input data, i.e., omissions. Despite recent efforts to minimize the appearance of these undesired divergences (Nie et al., 2019; Dušek and Kasner, 2020), further research is needed to deal properly with them when building neural models for D2T systems.

The first step to minimize hallucination and omission on neural models is to detect them. The task of detection requires checking for each generated text if its content matches with the input provided to the model. But it depends on the task for which the model has been designed. The input of an NLG system can be provided in different forms (e.g., structured meaning representation, images, tabular data, or text). In this paper, we focus our research on the detection of hallucinations and omissions when performing a D2T task in which the input is tabular data. Accordingly, we must

analyze the content of a generated text, extract its meaning, and then check the consistency or divergence with respect to the data table that was provided as input to the generation system.

Performing this task manually is tough and costly, in terms of time and human resources. Nevertheless, due to the variety and diversity of the texts generated by neural models, sometimes a fully automatic detection does not work properly because of context dependencies, ambiguity, or domain-specific language that only humans can understand. Thus, in this work, we first perform an automatic detection of divergences with our detector, and then a human expert analysis over the detected hallucinations. Notice that, the tasks to be carried out here are aligned with the error annotation and error-based evaluation proposed by (Thomson and Reiter, 2020).

Our focus is on an end-to-end D2T system for meteorology. Let us introduce an example. We can see in Fig. 1 a case of hallucinated content. The generated text refers to “hail” although there is no evidence of hailstone anywhere in the data. Thus, the generated text includes content which is not present in the input data. However, when we asked a meteorologist to rate the severity of this hallucination, he rated it as acceptable because “when there is rainy weather in the whole region there is a chance for occasional hail in some locations”. This type of cases highlights the importance of considering explicit domain knowledge, something that neural models are not able to achieve by themselves, since they only operate with the provided data.

The main contributions in this work are:

1. A new available Spanish dataset for D2T, including a clean corpus of meteorological texts: *MeteoGalicia-ES*¹. It is made up of 3,033 state-of-the-sky descriptions written by meteorologists, along with the corresponding tabular data for each described situation.
2. An adaption of a Transformer-based model to generate weather descriptions in Spanish from the tabular data in the *MeteoGalicia-ES* dataset.
3. An expert analysis of hallucinations over a set of divergences previously identified with the proposed detector of D2T divergences.

¹<https://gitlab.citius.usc.es/gsi-nlg/meteogalicia-es>

Input data table:

Zone	Morning	Afternoon	Night
Mariña Oriental	weak showers	showers	weak showers
Mariña Occidental	weak showers	showers	weak showers
...
Deza	weak showers	showers	sunny intervals

Generated text: The skies are expected to be cloudy with intermittent showers, occasionally stormy and accompanied by **hail**, more frequent in the morning.

Reference text: Skies will remain partly cloudy with showers, more frequent in the west.

Figure 1: Illustrative example of divergence between input data and output text of a neural D2T system. The hallucinated content is highlighted in red.²

The rest of the manuscript is organized as follows. Section 2 introduces related work. Section 3 presents the new dataset. Section 4 presents the proposal of a neural D2T system for the use case under consideration. Section 5 presents the approach for detecting divergences between input and output, along with the domain-expert analysis over hallucinations. Finally, Section 6 concludes the paper with some final remarks and points out future work.

2 Background

2.1 Data-to-text

One of the most popular and complete books on NLG, centered on D2T, was published by Reiter and Dale (1997). But, since the publication of this pioneering book, new methods have been developed in the field of NLG and, in particular, in the D2T subfield. Nowadays, rule-based or template-based systems tend to be replaced by deep learning models derived from the Machine Learning field, as described by Gatt and Krahmer (2018). Traditionally, NLG had to address, at least, two main tasks (usually addressed independently): the content selection, i.e., selecting the appropriate pieces of information to include in the final narrative; and the surface realization, i.e., communicating the selected information in the right format. However, end-to-end models are capable of addressing the whole generation pipeline at once, thus generating more complex outputs than traditional models while learning lexical and syntactic richness from large corpus and associated datasets.

²The original texts were in Spanish. We provide in the Figure the English translation.

Moreover, the development of the attention mechanism and the Transformer architecture (Vaswani et al., 2017) revolutionized both NLP and NLG fields. Even though, initially, Transformer models were used mainly for NLP tasks (e.g., question-answering or summarization) and text-to-text generation, their use in the context of D2T has also increased during last years (Devlin et al., 2019; Radford et al., 2019; Raffel et al., 2019).

In this paper, we focus on a subtask of D2T, named table-to-text, which aims to produce textual descriptions from an input in the form of structured tabular data. Recently, some end-to-end models were proposed to accomplish this task. For example, Puduppully et al. (2019) designed and developed a neural model which creates entity-specific representations, avoiding treating entities as simple vocabulary tokens. In addition, Gong et al. (2019a) and Rebuffel et al. (2019) proposed the use of hierarchical models in order to pay attention to different dimensions of tabular data. The former focuses on row, column, and time dimensions, while the latter encodes the input data at both element and structure level.

It is worth noting that even if it is well known that end-to-end models need large datasets to be properly trained, in the case of the table-to-text task there is still a lack of public datasets including human-written texts paired with tabular data. Indeed, some of the most popular datasets used to accomplish this task are from the sports domain, such as ROTOWIRE (Wiseman et al., 2017) and MLB (Puduppully et al., 2019) which include human-written summaries aligned with box-score data. In addition, if we look for open-domain datasets, we can find datasets like ToTTo (Parikh et al., 2020) and WIKIBIO (Lebret et al., 2016), both including tabular information and texts extracted from the Wikipedia.

It is worth noting that all the mentioned datasets are in English, and there is an evident lack of D2T resources in other languages. Here, one of our contributions is providing the NLG community with a new Spanish dataset composed of meteorological tabular data, aligned with textual descriptions made by experts in the field.

2.2 Hallucination and omission in D2T

Although end-to-end NLG models usually produce text which is characterized by fluency and natural-

ness, the fidelity to data of such text is sometimes arguable. Some generated texts mention false information, information that is not in the data, or simply ignore some relevant data. These phenomena, in many cases, are not acceptable (e.g., generation of medical or financial reports) and in many others make the text simply unpleasant or useless for the user (e.g., a virtual hotel advisor that gives you false information or omits good deals), which jeopardizes trust and credibility.

Accordingly, there has been an effort to propose novel methods to detect and minimize negative effects associated to hallucinations and/or omissions, and that way contributing to a more responsible NLP. Some studies showed how semantic noise in training data may lead neural models to divergence between input and output, either in the form of omissions or hallucinations (Dušek et al., 2019). Thus, some authors (Wang, 2019; Nie et al., 2019) proposed to reduce noise in training data with the aim of producing more consistent texts, while maintaining good fluency. In addition, Rebuffel et al. (2021) opted for enhancing the neural models instead of cleaning the datasets: they proposed the use of a decoder to leverage word-level labels and to learn relevant parts of each data instance. In the context of text-to-text summarization, Feijo and Moreira (2021) proposed first the creation of different “views” of the source text and then the selection of those candidate summaries which were more faithful to the source.

Notice that, all the proposals mentioned above are aimed to reducing the apparition of divergence between input and output for a given dataset and a specific model. Nevertheless, if we want to address the problem in a general way, we must address first the detection and classification of divergences and then, we may select the right way to deal properly with each case of hallucination or omission. Accordingly, Maynez et al. (2020) carried out a thorough analysis on different types of hallucination in the context of summarization. Human annotators read multiple summaries and identified both intrinsic (i.e., manipulating the information obtained from the input) and extrinsic (i.e., adding information beyond the one directly inferred from the input) hallucinations. This analysis reveals the dimension of the problem, which affects not only the summarization but also all tasks related to end-to-end NLG neural models.

In addition, Dušek and Kasner (2020) presented

a metric for evaluating D2T semantic accuracy based on Natural Language Inference. This metric detects both hallucinations and omissions automatically, but it is only for tasks where no content selection is required. Furthermore, there are some cases in which an automatic metric is not faithful enough to analyze the goodness of texts (e.g., context dependencies or domain-specific vocabulary) and complementary human evaluation is required.

In this paper, we make an expert analysis over different types of divergences detected by our automatic detector. First, the detector identifies both hallucinations and omissions from the output of an end-to-end D2T Transformer-based neural model. Then, an expert meteorologist analyzes the severity of the different types of hallucinations previously detected, and remarks the importance of considering contextual commonsense knowledge as part of the generation process.

3 The MeteoGalicia-ES Dataset

Weather forecasting is a popular topic in the D2T research field. There are some well-known datasets. For example, SUMTIME (Sripada et al., 2002) and WEATHERGOV (Liang et al., 2009). Here, we introduce a new dataset (MeteoGalicia-ES) which is made up of 3,033 records of meteorological tabular data along with handwritten textual descriptions in Spanish. Notice that, the dataset comprises real data and texts written by meteorologists. It was provided by MeteoGalicia, the Official Meteorological Agency of Galicia³.

3.1 Data tables

The data contained in MeteoGalicia-ES represent the state-of-the-sky by categorical values (e.g., “sunny”, “clouds”, “rain”, “fog”, etc.). The data provided in the dataset is organized in the form of different instances, each one composed by a table divided into 4 columns and 32 rows. The first column indicates the geographical zone of interest in Galicia, which covers a group of councils, while the remaining columns contain a value for each period of the day (morning, afternoon and night). This way, we have 3 state-of-the-sky values for each of the different 32 zones in Galicia, i.e., 96 (3 × 32) values per table.

All in all, in agreement with MeteoGalicia’s Style Guide, there are 20 different possible values for the state-of-the-sky, such as “rainy”, “high

clouds”, “clear”, etc. Unfortunately, being real data, the distribution of these data values is not homogeneous in the dataset. Therefore, in order to provide readers with useful and meaningful statistics, we have grouped the 20 possible values into 6 main categories regarding similar weather events, which are ranked in terms of their coverage of the dataset. We considered only those events which are in MeteoGalicia’s Style Guide. Each one of these events is represented in maps by a single specific icon, while textual descriptions admit some variety in the form of a list of admitted synonymous.

1. **Cloud:** it contains the four events that involve any type of clouds: (1.1) “sunny intervals”, (1.2) “clouds”, (1.3) “high clouds”, (1.4) “cloudy with sunny spells”, and (1.5) “covered”. This is by far the main category which covers a 47.3% of the data values, i.e., nearly the half of the cases in the dataset are related with events regarding clouds.
2. **Rain:** it contains the six events that involve water dropping: (2.1) “weak rains”, (2.2) “showers”, (2.3) “rain”, (2.4) “weak showers”, (2.5) “drizzle” and (2.6) “cloudy with showers”. This category is associated with the 27.6% of cases in the dataset.
3. **Clear:** it contains only the value (3.1) “clear”, i.e., what applies when there is no more than sun in the sky. This category represents the 21.5% of cases in the dataset.
4. **Snow:** it contains four events which involve frozen water: (4.1) “snow showers”, (4.2) “snow”, (4.3) “hail” and (4.4) “sleet”. This category only covers the 1.7% of cases in the dataset.
5. **Fog:** it contains three events which involve visibility reduction: (5.1) “fog”, (5.2) “fog banks” and (5.3) “mist”. Only 1.6% of cases are in this category.
6. **Storm:** it contains only the value (6.1) “storm”, i.e., what applies when electrical events (thunder and lightning) appear in the sky. This is by far the most underrepresented category, with only 0.3% of cases.

It is also worth noting that some state-of-the-sky values do not appear repeatedly in the same

³<https://www.meteogalicia.gal>

data instance. For example, the “snow” value appears only in specific zones in the region, i.e., in a particular cell of the data table. In addition, if we only take into account the single apparitions of the values in each data table (i.e., if a value appears more than once in an instance, it counts only as one) the computed statistics are quite different from the introduced above. In the 93.74% of data tables, there is at least one reference to the **Cloud** category. This means that almost all the meteorological situations from the dataset include weather phenomena involving clouds. The second most common category is **Rain**, with the 68.84% of the records referring to some rain phenomena. In addition, the **Clear** category covers nearly the half of the tables (47.25%) and the **Fog** category covers the 40.45% of tables. **Snow** and **Storm** are the most underrepresented categories, covering 14.41% and 8.41% of tables, respectively.

As we can see, the weather categories in MeteoGalicia-ES are unbalanced, some categories are overrepresented (e.g., **Cloud** and **Rain**) while others (e.g., **Snow** and **Storm**) are underrepresented. This is due to the fact that we are dealing with real data which were collected from 2010 to 2020, so they provide us with a complete picture of the weather in the Galician region during these period.

3.2 Texts

Associated to each data table, there is a textual description written by a meteorologist. All in all, there are 3,033 short meteorological descriptions of the state-of-the-sky made by experts in the field. Each description was cleaned and cured, correcting common punctuation or spelling typos. The length of the texts is variable, from a minimum of 25 characters until a maximum of 557 characters. The average length of the descriptions is 186 characters, while the standard deviation is 71.

We also made a deeper analysis of the collected texts, taking into account the type of textual references that they include. We considered both value references and spatial references. Value references match a state-of-the-sky value from the mentioned in section 3.1 (e.g., “fogs”, “rain”, “hail”, etc.), while spatial references determine where a weather phenomenon takes place (e.g., “coast” vs “inland”, or “north” vs “south”). In order to detect these two types of reference, we performed different searching methods based on the MeteoGalicia’s

Style Guide. This guide contains the vocabulary which must be used to refer to each weather phenomenon, and also the correct spatial references to name each zone in the map. This way, we created a dictionary with all potential expressions used by meteorologists when referring to zones and state-of-the-sky values. As a result of our analysis, we found out that in each text from the corpus, there are on average 2.53 value references and 1.66 spatial references. As expected, since texts describe the state-of-the-sky situation of a day in Galicia, we have more value references than spatial ones. Having between two and three value references per text means that data tables and descriptions are well aligned. It must be also highlighted the presence of above 1.5 spatial references in each text, which denotes the importance of this type of expressions in weather descriptions.

Additionally, we performed an analysis over temporal references, i.e., expressions that determine when a phenomenon occurs. In this case, we could not trust the vocabulary established by the MeteoGalicia’s Style Guide because it does not say anything about temporal references. Therefore, we performed a preliminary ad-hoc search of simple expressions (e.g, morning, afternoon, or night). Following this naive approach, we discovered on average about 1.07 temporal references in each text. Taking into account that we have probably overlooked some temporal expressions and therefore underestimated their presence in the dataset, we think they are likely to play a relevant role in the detection of hallucinations and/or omissions, and we will address this important issue in future work.

4 Data-to-text generation

This section describes an end-to-end D2T neural model which is trained with the MeteoGalicia-ES dataset previously introduced. Instead of designing a D2T system from scratch, we have reused the architecture of an existing Transformer-based model (Obeid and Hoque, 2020) which is carefully modified to be effective in our use case: generation of textual descriptions from tabular meteorological data. Nevertheless, it is worth noting that for the purpose of this paper, we do not need building the best (or a very good) D2T system for the given use case. This is because our ultimate goal, which will be carefully addressed in the next section, is testing an approach for automated detection of hallucinations and omissions previous to a care-

ful expert analysis over the detected cases. In this context, having a D2T system which performed perfectly free of divergences between inputs and outputs would make our experiment useless.

In the rest of this section, we first describe the Transformer-based architecture that is taken as base model. Then, we go in detail about how it has been reused, enhanced, trained and tested with the MeteoGalicia-ES dataset in order to generate weather forecasts in Spanish.

4.1 Base model

We took as starting point the Chart-to-text model (Obeid and Hoque, 2020). Given a chart and its title, this model describes the data embedded and depicted in the chart. Chart-to-text extends another previous Transformer-based D2T model (Gong et al., 2019b) in the following way: (i) Chart-to-text passes from input rows to input records, as a result it facilitates the addition of contextual information to the D2T system; (ii) Chart-to-text reintroduces positional embeddings as defined in the pioneering Transformer-based models for machine translation (Vaswani et al., 2017); and (iii) Chart-to-text can be fed with both numerical and categorical data values. These extensions are well aligned with our purposes because (i) we deal with more than the four values per tuple which were allowed by the original model; (ii) weather forecasting requires dealing with ordered/temporal relationships; and (iii) we have categorical values, such as the state-of-the-sky for each zone in Galicia (see the categories that we introduced in Section 3).

Additionally, the Chart-to-text base model includes a pre-processing stage initially thought for minimizing overfitting of the model but which can be seen as a very naive way for minimizing hallucinations, as we will see in the next section. More precisely, before training the model, the gold summaries in the corpus, i.e., original reference summaries, are pre-processed as follows: each token that refers to a value included either in the data table or in the chart title is replaced by a predefined label. This way, the model learns to generate more generic template-based summaries, i.e., non-value-dependent texts.

4.2 Our approach

Due to the nature of the data in MeteoGalicia-ES, we had to carry out several modifications on the base model with the aim of making it operative. First, our corpus is in Spanish while the base model

was thought for being trained with a corpus in English. Second, the MeteoGalicia-ES dataset comes from the specific field of meteorology, while the base model was aimed for describing generic charts from any field. In the rest of this section we explain in detail, step by step, how we have recycled and extended the base model.

4.2.1 Input data and pre-processing

Since we are dealing with tabular data, we maintain the base format. In the base model, each chart came with a data table and a brief title, which was taken into account when generating the descriptions. In our case, each table comprises all available meteorological data for one given day, i.e., it includes categorical values associated to the state-of-the-sky for each zone in Galicia and period of the day (morning, afternoon, night). We also added a generic title (“Weather forecast of a day in Galicia, by period of the day”) to each data table. This way, the D2T system can extract relevant tokens from the title during text generation. Notice that, each data table and title have the textual description in Spanish attached, which was handwritten by a meteorologist. Therefore, in the data pre-processing stage, our model had to be pre-trained to identify relevant tokens in Spanish before being ready to use them properly in the text generation stage. Similarly to Chart-to-text, we applied named entity recognition (Manning et al., 2014) to MeteoGalicia-ES with the aim of extracting important information from the given descriptions and titles associated to each data table.

4.2.2 Training and validation

Regarding the training and validation stages, we reused the architecture of the base neural model (Obeid and Hoque, 2020) with some variations in the parameters. We first randomized all the MeteoGalicia-ES instances and then used the 70% of them for training, 15% for validation and 15% for testing. The model was trained for 10 epochs with an epoch size of 1000, a dropout rate of 0.1, using 1 encoder layer, 6 decoder layers, embedding size of 512, batch size of 6, and beam size of 4. We used the hyperparameter values recommended by Chart-to-text without additional hyperparameter search. The model was trained on a GeForce RTX-2080 machine. The whole training took around 30 minutes. Once the model was trained, it was able to generate templates, i.e., texts with some gaps to fill with values from the input data.

4.2.3 Testing and post-processing

In the testing stage, the pre-trained model was provided only with the testing tabular data, and it was able to generate the final texts by filling in the previously generated templates. In the base model, each label in a gold template referred directly to a single value in the data table or to a single word in the title. Accordingly, filling in the given templates was straightforward. In our case, labels in templates are directly replaced by the given values only if the labels refer to values in the title. Otherwise, the BETO model (Cañete et al., 2020), pre-trained on a big Spanish corpus (Cañete, 2019), is applied to fill in the gap. This model is a Spanish version of the BERT model (Devlin et al., 2019) which replaces each label referring to tabular data with the best word from a set of candidates which includes the values in the corresponding category of data values. This way we improve naturalness while ensuring that gaps in templates are filled only with words that match the context of the sentence, thus minimizing typos as well as syntactic errors in surface realization. Finally, we run a post-processing step for polishing the generated texts and fixing some writing and/or concordance errors (e.g., fixing the use of capital letters after a full stop, verifying concordance of words in gender and number, removing repetitions of words, etc.).

5 Hallucination and Omission detector

This section first introduces and then validates our proposal for detecting hallucinations and omissions in texts generated by neural D2T systems. While the proposed approach is generic, it is validated in the meteorology use case we are considering.

The divergence detector is a software application composed of two independent parts, one for detecting each type of divergence. On the one hand, the omission detection part works as follows: it looks first at the table with input data values (i.e., identifies all state-of-the-sky values which apply to the case under consideration) and then checks if all these values are mentioned in the generated text. The detector counts as omission each value which is in the input data but is not explicitly referred to in the output text. On the other hand, the hallucination detection part follows the other way round. It looks first to the output text, identifies all data values which are mentioned in the text, and then checks if they are also included in the related input data. The detector counts as hallucination each

value which is mentioned in the output text but is not present in the input data.

It is worth noting that the current detector only looks for exact values, i.e., synonyms are not taken into consideration during the detection stage, what we are aware is a limitation of the present proposal to be addressed as future work. With the aim of evaluating the goodness of the proposal, we have validated the divergence detector with all the 272 unseen cases in the test set which was introduced in the previous section.

5.1 Reported omissions

Making use of our detector, we found that the number of omissions detected was very high. We identified omissions in 160 out of 272 texts (58%). This result shows how frequent omissions are in texts generated by neural models. However, further research is needed to assess how many of those omissions are admissible, and then refining accordingly our detector with the aim of reporting only those omissions that are more likely to be negatively perceived by humans. Indeed, omissions are naturally used by humans (as well as by traditional non-neural NLG systems) and they may be sometimes well appreciated because of producing shorter texts which only mention the most relevant pieces of information (as traditional NLG systems do thanks to the explicit stage of content determination).

For example, the data table associated to a given case includes the value “high clouds” while the output text refers to “open skies will prevail”. Formally speaking, this case counts as an omission because “high clouds” are not explicitly mentioned in the text. However, it should not because the generated text is considered valid by the meteorologist since it makes sense and conveys the correct information. This kind of cases are easily evaluated by humans, but really hard to be identified correctly by an automatic detector.

In order to identify which omissions could be admissible for humans and therefore should not be reported as unacceptable by our detector, we asked a meteorologist to analyze in detail a group of randomly selected cases among the detected omissions. He confirmed that many of them were admissible because in the context of meteorology missing some information is not so severe as it may be in other application domains. In fact, in some cases, the meteorologist preferred certain omission to the exhaustive verbalization of all the values in

the data table what could lead to a long, verbose, repetitive and less natural text.

It is worth noting that our results are in agreement with those reported by related work in which a similar analysis was done. For example, [Dušek et al. \(2019\)](#) and [Nie et al. \(2019\)](#) also reported many omissions when analyzing the content coverage of texts generated by neural models. They also noted that forcing the model to verbalize all slots during training leads to fewer omissions but at the cost of producing longer texts.

5.2 Reported hallucinations

The texts generated by our model describe meteorological situations in a geographical region, but the handwritten reference texts sometimes describe the state-of-the-sky of specific zones inside the whole Galician region, e.g., “Skies will be cloudy in the Atlantic coast”. Considering this, if our model generates a text in which a state-of-the-sky value is associated to a wrong zone (e.g., following the previous example, there are no clouds in the Atlantic coast), it must be considered also a case of hallucination.

Accordingly, we analyzed two different levels of hallucination: basic hallucinations and spatial hallucinations. The former are cases in which the model generation adds information not directly inferable from the input, i.e., extrinsic hallucinations, while the latter are generations in which the model manipulates geographical information inferred from the input (it could be considered as an intrinsic domain-specific hallucination).

Once again, we followed the MeteoGalicia’s Style Guide, with the aim of identifying the list of admissible spatial references along with their related locations in the map. As a result, we identified 48 different reference expressions that meteorologists may use to refer to different geographical zones in Galicia.

The detector identified 35 basic hallucinations and 11 spatial hallucinations out of all the 272 texts under study. In order to assess the goodness of the detector and to determine if all reported hallucinations were really worthy to note, we asked once again the assistance of a meteorologist. He rated the degree of relevance of each detected hallucination in a 3-points Likert scale (admissible, partially admissible, inadmissible). Surprisingly, 12 (10 basic and 2 spatial hallucinations) out of all the 46 detected hallucinations were deemed as admissible.

Formally speaking, all these 12 cases were hallucinations (i.e., the state-of-the-sky values mentioned in the output text were not present in the input data) but, according to the meteorologist’s background and in agreement with contextual information and commonsense reasoning, they were admissible.

Figure 1 depicted an example of admissible hallucination. Even if according to the strict data checking done by our detector this is a case of hallucination, the meteorologist rated it as admissible due to the observed situation in the whole region, which according to his experience justifies a very high possibility of hail. It is also worthy to note that in four of the hallucinations rated as admissible by the meteorologist, the reference texts in the corpus also mentioned some values which were not in the data. For example, in one of the cases, both reference and model texts mention “storm with hail” while in the associated data there are only “storm” values. This suggests that it may be a good thing to use the detector as part of the pre-processing stage for automatically identifying and fixing similar cases that are likely to be included in the training set. We will address this challenging task in the near future.

6 Final Remarks and Future Work

In this paper, we first introduced a new dataset (MeteoGalicia-ES) for D2T in the application domain of meteorology. Then, we reused and adapted a neural D2T system to generate weather descriptions from MeteoGalicia-ES. Finally, we described an approach to automatically detect and validate hallucinations and/or omissions in the texts generated by the D2T system previously trained.

In the light of the reported results, we can draw the following important remarks. First, neural D2T systems, after being trained with large-scale datasets, can generate natural and fluid texts, but more often than not the generated texts provide unfaithful information or inconsistencies with respect to the input data, mainly in the form of omissions and/or hallucinations. In our specific use case, we detected more omissions than hallucinations, but in general hallucinations were more negatively perceived and deemed as misleading by the meteorologist who assisted us in the validation stage. Notice that the observed divergence between input and output in some controversial cases is likely to be due to the lack of ability of the designed D2T system to deal with contextual information and com-

nonsense reasoning as humans naturally do. In addition, we must take into account that in practice, meteorologists rely on contextual information and commonsense reasoning beyond input data when writing weather forecasts. Current neural D2T systems can not capture such a general knowledge because they are only guided by the given training data. This means that for truly complex tasks, where either omissions or hallucinations may be critical, neural models have to be endowed and integrated with other knowledge sources different from data, if we want them to achieve high quality automatically generated texts which are as correct as expert-made ones.

Last but not least, the high level of naturalness and fluidity that neural D2T systems usually achieve may raise too high expectations in end users, who may be frustrated when discovering some misleading pieces of information. We claim that providing users with the generated texts and the findings of our detector contributes to lowering expectations, in the sense that we make explicit limitations and undesired behaviors of the underlying D2T system. This way, we contribute to a more responsible NLP.

As future work, we plan in the midterm to enrich our neural D2T system with a knowledge base including meteorological facts (regarding both spatial and temporal references) but also in the long-term with temporal knowledge. As a result, we expect to improve both text generation and hallucination/omission detection. Moreover, we will go deeper with understanding how classical NLG approaches for content determination can help to identify relevant omissions.

Acknowledgments

Jose Maria Alonso-Moral is a *Ramón y Cajal* Researcher (RYC-2016-19802). This research was funded by the Spanish Ministry for Science, Innovation and Universities (grants PID2020-112623GB-I00, and PDC2021-121072-C21) and the Galician Ministry of Culture, Education, Professional Training and University (grants ED431F 2018/02, ED431C 2018/29, ED431G/08, ED431G2019/04, ED431C2022/19). All grants were co-funded by the European Regional Development Fund (ERDF/FEDER program).

References

- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). *CoRR*, abs/2005.14165.
- José Cañete. 2019. [Compilation of large spanish unannotated corpora](#). Zenodo.
- José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. 2020. [Spanish pre-trained bert model and evaluation data](#). In *PMLADC at ICLR 2020*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Ondřej Dušek, David M. Howcroft, and Verena Rieser. 2019. [Semantic noise matters for neural natural language generation](#). In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 421–426, Tokyo, Japan. Association for Computational Linguistics.
- Ondřej Dušek and Zdeněk Kasner. 2020. [Evaluating semantic accuracy of data-to-text generation with natural language inference](#). In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 131–137, Dublin, Ireland. Association for Computational Linguistics.
- Diego Feijo and Viviane Moreira. 2021. [Improving abstractive summarization of legal rulings through textual entailment](#). *Artificial Intelligence and Law*.
- Albert Gatt and Emiel Kraemer. 2018. [Survey of the state of the art in natural language generation: Core tasks, applications and evaluation](#). *Journal of Artificial Intelligence Research*, 61(1):65–170.
- Heng Gong, Xiaocheng Feng, Bing Qin, and Ting Liu. 2019a. [Table-to-text generation with effective hierarchical encoder on three dimensions \(row, column and time\)](#). In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3143–3152, Hong Kong, China. Association for Computational Linguistics.

- Li Gong, Josep Crego, and Jean Senellart. 2019b. [Enhanced transformer model for data-to-text generation](#). In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 148–156, Hong Kong. Association for Computational Linguistics.
- Rémi Lebret, David Grangier, and Michael Auli. 2016. [Neural Text Generation from Structured Data with Application to the Biography Domain](#). In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.
- Percy Liang, Michael Jordan, and Dan Klein. 2009. [Learning semantic correspondences with less supervision](#). In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 91–99, Suntec, Singapore. Association for Computational Linguistics.
- Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. [The Stanford CoreNLP natural language processing toolkit](#). In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.
- Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. [On faithfulness and factuality in abstractive summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.
- Feng Nie, Jin-Ge Yao, Jinpeng Wang, Rong Pan, and Chin-Yew Lin. 2019. [A simple recipe towards reducing hallucination in neural surface realisation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2673–2679, Florence, Italy. Association for Computational Linguistics.
- Jason Obeid and Enamul Hoque. 2020. [Chart-to-text: Generating natural language descriptions for charts by adapting the transformer model](#). *CoRR*, abs/2010.09142.
- Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. [ToTTo: A controlled table-to-text generation dataset](#). In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186, Online. Association for Computational Linguistics.
- Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. [Data-to-text generation with entity modeling](#). *CoRR*, abs/1906.03221.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. [Language models are unsupervised multitask learners](#).
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *CoRR*, abs/1910.10683.
- Clément Rebuffel, Marco Roberti, Laure Soulier, Geoffrey Scoutheeten, Rossella Cancelliere, and Patrick Gallinari. 2021. [Controlling hallucinations at word level in data-to-text generation](#). *CoRR*, abs/2102.02810.
- Clément Rebuffel, Laure Soulier, Geoffrey Scoutheeten, and Patrick Gallinari. 2019. [A hierarchical model for data-to-text generation](#). *CoRR*, abs/1912.10011.
- Ehud Reiter and Robert Dale. 1997. [Building applied natural language generation systems](#). *Natural Language Engineering*, 3(1):57–87.
- Somayajulu Sripada, Ehud Reiter, Jim Hunter, and Jin Yu. 2002. [Sumtime-meteo: Parallel corpus of naturally occurring forecast texts and weather data](#). *Computing Science Department, University of Aberdeen, Aberdeen, Scotland, Tech. Rep. AUCS/TR0201*.
- Craig Thomson and Ehud Reiter. 2020. [A gold standard methodology for evaluating accuracy in data-to-text systems](#). In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 158–168, Dublin, Ireland. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Hongmin Wang. 2019. [Revisiting challenges in data-to-text generation with fact grounding](#). In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 311–322, Tokyo, Japan. Association for Computational Linguistics.
- Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. [Challenges in data-to-document generation](#). In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Amortized Noisy Channel Neural Machine Translation

Richard Yuanzhe Pang
New York University
yzpang@nyu.edu

He He
New York University

Kyunghyun Cho
New York University
Genetech
CIFAR Fellow

Abstract

Noisy channel models have been especially effective in neural machine translation (NMT). However, recent approaches like “beam search and rerank” (BSR) incur significant computation overhead during inference, making real-world application infeasible. We aim to study if it is possible to build an amortized noisy channel NMT model such that when we do greedy decoding during inference, the translation accuracy matches that of BSR in terms of reward (based on the source-to-target log probability and the target-to-source log probability) and quality (based on BLEU and BLEURT). We attempt three approaches to train the new model: knowledge distillation, 1-step-deviation imitation learning, and Q learning. The first approach obtains the noisy channel signal from a pseudo-corpus, and the latter two approaches aim to optimize toward a noisy-channel MT reward directly. For all three approaches, the generated translations fail to achieve rewards comparable to BSR, but the translation quality approximated by BLEU and BLEURT is similar to the quality of BSR-produced translations. Additionally, all three approaches speed up inference by 1–2 orders of magnitude.

1 Introduction

Noisy channel models have been traditionally used in many tasks, including speech recognition (Jelinek, 1997), spelling correction (Brill and Moore, 2000), question answering (Echihabi and Marcu, 2003), and statistical machine translation (Koehn et al., 2003). In machine translation (MT), the probability of the source sentence conditioned on the target-language generation is taken into account when generating a translation. In modern neural machine translation (NMT), the noisy channel approach is successful and often indispensable in many recent top-performing machine translation systems (Yee et al., 2019; Ng et al., 2019; Chen et al., 2020; Yu et al., 2020; Tran et al., 2021).

One widely used approach of noisy channel NMT is “beam search and rerank” (BSR). Assume a trained forward translator and a trained reverse translator,¹ BSR decoding consists of two steps: first, decode using beam search with a large beam size from the forward translation model and store the entire beam; second, rerank the beam using a reward which is the sum of the forward translation log probability and the reverse log probability. This approach incurs significant computational overhead, given the need to decode a large beam (usually with beam size 50–100) from the forward translator and the need to feed the large beam through the reverse translator. The computational cost is especially problematic if the practitioner has a large volume of translation requests, or if the system is mobile-based and requires offline translation.

We thus aim to learn a separate neural network with an identical architecture as the forward translator such that at inference time, when we do *greedy decoding* using this new network, we investigate how much translation accuracy would be sacrificed. Specifically, we investigate how forward/reverse rewards of the translations as well as the translation quality (approximated by BLEU and BLEURT) would compare to those of BSR-generated translations.²

The paper explores three approaches, with increasingly more exploration when optimizing the reward. (1) Knowledge distillation (KD) from a pseudo-training-corpus generated by BSR: we can treat the BSR-generated corpus as the oracle, and KD can be interpreted as behavioral cloning. (2) a 1-step-deviation imitation learning strategy (IL) where given a fixed sequence of target-language

¹Forward: from the source language to the target language; reverse: from the target language to the source language.

²Although we need time to train the separate network, at inference time and during the actual large-scale user-facing deployment, we would be able massively cut down computational cost in the long run. In this paper, we aim to investigate the accuracy of such decoded translations.

tokens, we adjust the next-time-step probability distribution over the vocabulary such that the resulting distribution minimizes an energy function used in BSR reranking, and (3) Q learning which explicitly learns the scoring function used in BSR reranking.

We experiment on three datasets (IWSLT’14 De-En, WMT’16 Ro-En, and WMT’14 De-En). Experimental results show that all three approaches speed up inference by 50–100 times. The approaches fail to achieve comparable rewards to BSR, but compared to the non-BSR baselines, the approaches achieve much higher reverse rewards (i.e., $\log p_r(\mathbf{x} | \mathbf{y})$ where p_r is the reverse translator) at the expense of forward rewards (i.e., $\log p_f(\mathbf{y} | \mathbf{x})$ where p_f is the forward translator). Meanwhile, the approaches achieve a translation quality (approximated by BLEU and BLEURT) that is comparable to that of BSR. In particular, IL’s BLEURT scores is significantly higher than those of beam search, across all three datasets; IL’s BLEURT scores are not significantly different from BSR’s scores, across three datasets.

2 Background

2.1 Neural Machine Translation

NMT systems usually model the distribution $p(\mathbf{y} | \mathbf{x})$ where $\mathbf{x} = (x_1, x_2, \dots, x_{T_s})$ is a source-language sequence and $\mathbf{y} = (y_1, y_2, \dots, y_T)$ is a target-language sequence. Most NMT systems use an autoregressive factorization:

$$\log p(\mathbf{y} | \mathbf{x}) = \sum_{t=1}^T \log p_\theta(y_t | \mathbf{y}_{<t}, \mathbf{x}),$$

where $\mathbf{y}_{<t} = (y_1, y_2, \dots, y_{t-1})$, and p_θ is parameterized with a neural network. At test-time, to decode a translation given a source sentence, greedy decoding and beam search are most commonly used. Both are approximate search methods to find the highest-scoring translations.

2.2 Beam Search and Rerank (BSR)

BSR has appeared in a number of top-performing models, including many winning submissions of the WMT competitions (Ng et al., 2019; Chen et al., 2020; Yu et al., 2020; Tran et al., 2021). The intuition of BSR is to take advantage of the reverse translator during decoding. Specifically, we do beam search with a large beam size b (usually 50–100) to obtain b candidate translations. Then, we

rerank the candidates using the scoring function:

$$\log p_f(\mathbf{y} | \mathbf{x}) + \gamma \log p_r(\mathbf{x} | \mathbf{y}) + \gamma' \log p_{lm}(\mathbf{y}),$$

where γ and γ' are tuned in $[0, 2]$. Without access to a language model trained on a huge target-language monolingual external corpus, if we use $\log p_f(\mathbf{y} | \mathbf{x}) + \gamma \log p_r(\mathbf{x} | \mathbf{y})$ as the ranking criteria, BSR also provides a significant performance gain. With a large beam size, this approach performs better than the “two-step beam search” approach (Yu et al., 2017; Yee et al., 2019).

3 Amortized Noisy-Channel NMT

One common problem with the above approaches is the inference-time computation overhead. If a translation system needs to translate a high volume of texts, then the test-time computational efficiency is crucial. Thus, our goal is to use a network to approximate such a noisy channel NMT system, while having the same inference-time computational cost as *greedily decoding from p_f* . Specifically, we want our translations to maximize the following objective:

$$R(\mathbf{x}, \mathbf{y}) = \log p_f(\mathbf{y} | \mathbf{x}) + \gamma \log p_r(\mathbf{x} | \mathbf{y}), \quad (1)$$

where $\gamma > 0$ is some fixed coefficient. Using the autoregressive factorization, the forward reward $\log p_f(\mathbf{y} | \mathbf{x})$ equals $\sum_{t=1}^{|\mathbf{y}|} \log p_f(y_t | \mathbf{y}_{<t}, \mathbf{x})$, and the reverse reward $\log p_r(\mathbf{x} | \mathbf{y})$ equals $\sum_{t=1}^{|\mathbf{x}|} \log p_r(x_t | \mathbf{x}_{<t}, \mathbf{y})$.

Goal: Investigating if greedily decoding from our new models leads to accurate translations.

Three approaches are shown in this section. We do greedy decoding from the obtained models, and we investigate the translation accuracy as follows. First, we examine if both the forward and reverse rewards of the translations are close to the forward and reverse rewards of the translations generated by BSR, respectively. Second, we examine the translation quality by checking if BLEU and BLEURT scores of our model’s translations are close to those of BSR-produced translations.

3.1 Approach 1: Knowledge Distillation (KD)

KD has been used to amortize beam search (Chen et al., 2018). It is also effective in NMT in general (Kim and Rush, 2016; Freitag et al., 2017; Tan et al., 2019; Tu et al., 2020). Here we adapt the simple KD for amortized noisy-channel decoding.

First, train a forward translator p_f and a reverse translator p_r using maximum likelihood estimation. Then, we do BSR on the entire training set to obtain the pseudo-corpus. In particular, we ignore the p_{lm} term given that it usually requires a giant language model, and the inclusion of the term is orthogonal to our goal of reducing inference time.³

Next, we train a separate “knowledge distilled” model p_{KD} on this new pseudo-corpus (i.e., with the original source-language sentences and the BSR-generated target-language sentences). This objective is equivalent to minimizing the KL-divergence between the distribution induced by the pseudo-corpus obtained through BSR and our model distribution.

At inference time, we greedily decode from p_{KD} .

3.2 Approach 2: 1-Step-Deviation Imitation Learning (IL)

Define a network A_ϕ such that it takes in the source sentence and a target-language prefix, and $A_\phi(\cdot | \mathbf{x}, \mathbf{y}_{<t})$ outputs a $|\mathcal{V}|$ -dimensional probability distribution corresponding to the t -th time-step. Moreover, A_ϕ and p_f have the same architecture. In autoregressive text generation, to learn A_ϕ such that it is close to an existing network p_θ , imitation learning seeks to optimize ϕ as follows:

$$\arg \min_{\phi} \mathbb{E}_{(\mathbf{x}, \mathbf{y}_{<t})} \mathcal{L}(A_\phi(\cdot | \mathbf{x}, \mathbf{y}_{<t}), p_\theta(\cdot | \mathbf{x}, \mathbf{y}_{<t})),$$

where one example of \mathcal{L} is the cross entropy.

Forward energy. Inspired by ENGINE (Tu et al., 2020), in the context of noisy channel NMT, define the forward sub-energy E_t^f , which is a function of ϕ , as follows:⁴

$$E_t^f(\mathbf{x}, \hat{\mathbf{y}}; \phi) = -A_\phi(\cdot | \mathbf{x}, \hat{\mathbf{y}}_{<t})^\top \log p_f(\cdot | A_\phi(\cdot | \mathbf{x}, \hat{\mathbf{y}}_{<1}), \dots, A_\phi(\cdot | \mathbf{x}, \hat{\mathbf{y}}_{<t}), \mathbf{x}).$$

Suppose we have a source sentence \mathbf{x} and a sequence of prefix distributions $\hat{\mathbf{y}}_{<1}, \dots, \hat{\mathbf{y}}_{<T}$. We

³Generating the pseudo-corpora can be paralleled. If the system is deployed in the real world, we argue that the amount of computation used to generate the pseudo-corpus is negligible, compared to the aggregate amount of computation for inference.

⁴If we compute $p_f(\cdot | y_1, y_2, \dots, y_{t-1}, \mathbf{x})$ where y_i ’s correspond to token IDs of a partial translation in the target language, then we would first look up the y_i -th row of the embedding matrix E_{emb} and use this row to embed y_i ; equivalently, we can use the product $\text{onehot}(y_i)^\top E_{emb}$ to embed y_i . In the case of E_t^f , the prefixes used in p_f are distributions instead of tokens. We can use $A_\phi(\cdot | \mathbf{x}, \hat{\mathbf{y}}_{<i})^\top E_{emb}$ to represent the i -th token embedding. The embedding strategy is similar for E_t^r below.

call $A_\phi(\cdot | \mathbf{x}, \hat{\mathbf{y}}_{<t})$ the t -th step distribution according to A_ϕ . Intuitively, given a source and a fixed sequence of prefixes, we learn A_ϕ such that the resulting t -th-step distribution matches the forward conditional probability (measured by p_f) which depends on the source \mathbf{x} and the prefix distributions.

Reverse energy. Next, we define the reverse sub-energy as follows:

$$E_t^r(\mathbf{x}, \hat{\mathbf{y}}; \phi) = -\text{onehot}(\mathbf{x}_t)^\top \log p_r(\cdot | \mathbf{x}_{<t}, A_\phi(\cdot | \mathbf{x}, \hat{\mathbf{y}}_{<1}), \dots, A_\phi(\cdot | \mathbf{x}, \hat{\mathbf{y}}_{<T})).$$

Intuitively, the one-hot distributions corresponding to the source words should match the reverse conditional probability (measured by p_r).

Trajectories. In the above equations, $\hat{\mathbf{y}} = (\hat{y}_1, \dots, \hat{y}_T)$. \hat{y}_t comes from two sources, with probability p and $1 - p$ for each minibatch during training (Section 4.2): (i) $\hat{y}_t = \arg \max_{v \in \mathcal{V}} A_\phi(\cdot | \mathbf{x}, \hat{\mathbf{y}}_{<t})$ and $\hat{y}_{<1} = \emptyset$; in other words, given that $A_\phi(\cdot | \mathbf{x}, \hat{\mathbf{y}}_{<t})$ is a probability distribution, we use the most likely token as \hat{y}_t . (ii) For the second source, let \hat{v}_t be the t -th token of the BSR-obtained sequence, so that we can expose our model to BSR-prefixes, which are the optimal prefixes.

Final objective. We train A_ϕ using the following objective:

$$\min_{\phi} \sum_{\mathbf{x}} \left[\sum_{t=1}^T E_t^f(\mathbf{x}, \hat{\mathbf{y}}; \phi) + \gamma \sum_{t'=1}^{|\mathbf{x}|} E_{t'}^r(\mathbf{x}, \hat{\mathbf{y}}; \phi) \right]$$

During inference, we greedily decode from A .

3.3 Approach 3: Q Learning

A well-motivated approach is to use Q learning (Watkins and Dayan, 1992; Sutton and Barto, 1998) to explicitly learn a reward function Q , with the goal that when we greedily decode from Q , the generations maximize the reward shown in Eq. (1).

Let us view machine translation as a sequential decision-making process. At time-step t , given a state $s_t = (\mathbf{y}_{<t}, \mathbf{x})$, a policy takes an action $a_t \in \mathcal{V}$, transits to the next state $s_{t+1} = (\mathbf{y}_{<(t+1)}, \mathbf{x})$ where $\mathbf{y}_{<(t+1)}$ equals $\mathbf{y}_{<t}$ concatenated with the action a_t , and receives a reward r_t .

3.3.1 Background on Q Learning

In Q learning, $Q^\pi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is a function such that $Q^\pi(s_t, a_t)$ produces the expected return after seeing state s_t , taking action a_t , and following

policy π ; i.e., $Q^\pi(s_t, a_t) = \mathbb{E}[\sum_{t'=t}^{\infty} r_{t'} | s_t, a_t, \pi]$ assuming discount factor 1. We further define $Q^* : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ to be the optimal action-value function: $Q^*(s_t, a_t) = \max_{\pi} \mathbb{E}[\sum_{t'=t}^{\infty} r_{t'} | s_t, a_t, \pi]$, which is the maximum return achievable by following any strategy *after* seeing a state s_t and taking an action a_t . In particular, Q^* solves the Bellman Equation (Sutton and Barto, 1998):

$$Q^*(s_t, a_t) = r_t + \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1}),$$

assuming discount factor 1 and given deterministic transition dynamics (in our machine translation scenario) after taking action a_t given state s_t .

Traditionally, the Q function is implemented as a matrix of size $|\mathcal{S}| \times |\mathcal{A}|$, which is intractable in the case of MT due to the combinatorial nature of the state space. We thus use function approximation to tackle this issue of intractability: we follow Mnih et al. (2015) and use a deep neural network trained with experience replay and target networks to approximate the Q learning.

Deep Q learning draws samples from a set of trajectories \mathcal{B} , and the neural network Q aims to predict Q^* by learning based on minimizing the following squared loss.

$$L(\phi) = \frac{1}{|\mathcal{B}|} \sum_{(s_t, a_t, s_{t+1}, r_t) \sim \text{Uniform}(\mathcal{B})} [(r_t + \max_{a_{t+1}} Q'(s_{t+1}, a_{t+1}) - Q(s_t, a_t))^2],$$

where ϕ is the parameter to Q , and Q' is a slightly old copy of Q .⁵

3.3.2 Q Learning for Amortized Noisy Channel MT

To model the noisy-channel NMT, given a target-language sequence \mathbf{y} and its length T , we have reward $\mathbf{r} = (r_1, \dots, r_T)$, where

$$r_t = \begin{cases} \log p_f(y_t | \mathbf{y}_{<t}, \mathbf{x}), & \text{if } t < T, \\ \log p_f(y_T | \mathbf{y}_{<T}, \mathbf{x}) + \gamma \cdot \log p_r(\mathbf{x} | \mathbf{y}), & \text{if } t = T. \end{cases} \quad (2)$$

We construct Q to have the same architecture as p_f without the final softmax layer.⁶ Q is trained using Algorithm 1 which is adapted from deep Q learning originally applied to Atari games (Mnih

⁵In other words, after a fixed number of optimization steps, we update Q' by Q .

⁶One corollary is that Q and p_f have the same number of parameters.

Algorithm 1: Q learning for amortized noisy channel NMT

Given p_f, p_r , and a parallel translation dataset \mathcal{D} .

while not converged do

 Collect training trajectories (§3.3), and sample a mini-batch \mathcal{B} .

 Compute target R_t : if $t < T$, then

$R_t = r_t + \max_{a_{t+1}} Q'_\phi(s_{t+1}, a_{t+1})$; if $t = T$, then $R_t = r_T$.

 Update ϕ (using gradient descent) by the objective $\arg \min_{\phi} [Q_\phi(s_t, a_t) - R_t]^2$.

 Update Q'_ϕ : $Q'_\phi \leftarrow Q_\phi$ every K steps.

end

et al., 2015), given that we aim to best leverage the existing off-policy trajectories from different sources. The full algorithm is shown in Algorithm 1.

In short, our algorithm says that given a trajectory $(\mathbf{x}, \mathbf{y}, \mathbf{r})$, at time-step $t < T$, we want the scalar $Q(s_t, a_t)$ to be close to the sum of the t -th step reward and the *most optimistic* future return, had we taken action a_t at time-step t . At time-step T , we want $Q(s_T, a_T) = Q((\mathbf{y}_{<T}, \mathbf{x}), \langle \text{eos} \rangle)$ to be close to r_T , as defined in Eq. (2).

To generate the t -th token at inference-time, we do greedy decoding as follows: $\hat{y}_t = \arg \max_{a_t \in \mathcal{V}} Q(s_t, a_t)$.

Trajectories. The off-policy Algorithm 1 requires trajectories, i.e., $(\mathbf{x}, \mathbf{y}, \mathbf{r})$ tuples. The trajectories come from two sources.

(1) Q -based trajectories. In this category, we have two ways of obtaining \mathbf{y} : (1a) Boltzmann exploration (Sutton, 1990)⁷ and (1b) greedy decoding based on Q . At the start of the optimization, however, most of the Q -generated sequences are very far from target sequences. The lack of high-reward sequences prevents Q learning from efficient optimization. Therefore, we also inject reasonably good trajectories from the beginning of training by utilizing both ground-truth sequences as well as p_f -based sequences. We thus need the next category

⁷Recall that at time-step t , $Q(s_t, a_t) \in \mathbb{R}$ for each $a_t \in \mathcal{V}$. Therefore, $Q(s_t, \cdot) \in \mathbb{R}^{|\mathcal{V}|}$. We turn the vector of real numbers to a categorical distribution by softmax with temperature γ_b . Then, the sequences in the trajectories are obtained by sampling from the aforementioned distribution. In practice, for each sequence, we use a temperature γ_b sampled from $\text{Uniform}(0, 1.5)$. One can think of this strategy as a variant of ϵ -greedy which is typically used in Q learning.

of sources.

(2) *p_f-based trajectories*. The target-language sequences are obtained by decoding using *p_f*; more details in Appendix A.1.⁸

4 Experimental Setup

4.1 Tasks and Models

We experiment on three translation tasks: IWSLT 2014 German to English (IWSLT’14 De-En; [Cettolo et al., 2014](#)) which has a small training set (train/dev/test size: 160,239/7,283/6,750), WMT 2016 Romanian to English (WMT’16 Ro-En; [Bojar et al., 2016](#)) which has a medium-sized training set (train/dev/test size: 608,319/1,999/1,999), and WMT 2014 German to English (WMT’14 De-En; [Bojar et al., 2014](#)) which has a moderately large training set (train/dev/test size: 4,500,966/3,000/3,003). Each of the transformer models (the *p_{KD}* in KD, the *A* in IL, the *Q* function in Q learning) has the same number of parameters as the original MLE-trained forward translator *p_f*. The model for IWSLT’14 De-En is the smallest, and the model for WMT’14 De-En is the largest. The detailed settings can be found in Appendix B. BLEU scores in this paper are computed with sacreBLEU ([Post, 2018](#)). BLEURT scores are computed using BLEURT-20-D12 ([Sellam et al., 2020](#)), a recent RemBERT-based checkpoint that achieves high human agreement. The models we experiment on are shown in Table 1.

4.2 Hyperparameters

The architecture and optimization details of *p_f* and *p_r* are shown in Appendix B. When training *p_f* and *p_r*, we validate the model performance after each epoch, and select the model that corresponds to the best dev set BLEU.

γ is the coefficient multiplied to the reverse reward, when computing the total reward in Eq. (1); γ and BSR beam size *b* are tuned on dev set BLEU using BSR. We choose $\gamma = 0.9$ and *b* = 100 for IWSLT’14 De-En; $\gamma = 0.5$ and *b* = 70 for WMT’16 Ro-En; $\gamma = 0.5$ and *b* = 50 WMT’14 De-En. See Appendix B for details.

For training the IL-based network, the learning rate is selected from $\{10^{-6}, 5 \times 10^{-6}, 10^{-5}, 3 \times 10^{-5}, 5 \times 10^{-5}\}$. We use weight decay of 10^{-4} .

⁸We have also experimented with gold-standard trajectories from the parallel translation dataset \mathcal{D} , but the inclusion of such trajectories do not lead to better rewards of *Q*-generated translations.

Dropout rate is selected from $\{0, 0.05, 0.1, 0.3\}$; we find that a dropout rate of 0 or 0.05 always works the best. We use a fixed max batch length (i.e., the max number of input tokens in a batch) of 4,096 tokens. The probability *p*, described in Section 3, is selected from $\{0, 0.1, 0.5, 0.9, 1\}$; we find that $p = 0.1$ or $p = 0.5$ usually works the best. We accumulate gradients and do gradient descent once every *k* steps for computational reasons. *k* is selected from $\{4, 8, 16\}$. We find that the IL approach relies on a good initialization, so we use *p_{KD/nc}* to initialize the new network.

For Q learning, the synchronization frequency *K* in Algorithm 1 is selected from $\{10, 20, 30, 50, 150\}$. The learning rate is tuned in $\{10^{-5}, 3 \times 10^{-5}, 5 \times 10^{-5}, 10^{-4}\}$. We use weight decay of 10^{-4} . Dropout rate is tuned in $\{0, 0.01, 0.05, 0.1\}$; we find that a dropout rate of 0 always works the best. We use a fixed max batch length 4096. We tune the number of steps per gradient update in $\{4, 8, 16\}$; a large number effectively increases the batch size. The ratio for different trajectories is described in Appendix A.1. Furthermore, we find that training *Q* with a small γ at the beginning stabilizes the training, so we first use $\gamma = 0.1$ and train till convergence, and then increase γ by 0.2 increment, and we reiterate the process until reaching the desired γ .

We use the Adam optimizer ([Kingma and Ba, 2014](#)) for all experiments. We cap the maximum length of the translation at $1.2T_s + 20$ during decoding, where *T_s* is the length of a source sentence. All implementation is based on fairseq ([Ott et al., 2019](#)). Each experiment is run on one NVIDIA RTX 8000 GPU.

5 Results

5.1 Preliminary Analysis

Inference speed. Using any of the three proposed approaches achieves a significant speedup, given that the three approaches all use greedy decoding. We quantify this speedup experimentally. During inference, we maximize the memory usage of a single NVIDIA RTX 8000 GPU by finding the largest batch length in the form of 2^k where *k* is a positive integer.⁹ In the IWSLT’14 De-En task, the inference speed (sequences per second) for BSR is 11. The speed for “greedy by *p_f*” is around 1050,

⁹Batch length means the number of tokens in a batch (instead of the number of sequences).

	IWSLT’14 De-En			WMT’16 Ro-En			WMT’14 De-En		
	b	fwd reward mean (std)	rvs reward mean (std)	b	fwd reward mean (std)	rvs reward mean (std)	b	fwd reward mean (std)	rvs reward mean (std)
p_f	1	-9.1 (7.7)	-35.4 (39.9)	1	-9.5 (11.5)	-41.0 (50.1)	1	-11.0 (6.3)	-31.5 (24.6)
p_f	5	-8.6 (7.0)	-34.2 (38.5)	5	-9.0 (8.5)	-40.2 (48.2)	7	-10.4 (5.5)	-29.9 (21.5)
BSR	100	-9.4 (6.8)	-25.7 (32.5)	70	-10.0 (6.0)	-29.7 (41.9)	50	-10.7 (5.3)	-23.6 (16.3)
KD	1	-13.8 (13.9)	-28.0 (32.7)	1	-17.2 (26.3)	-35.4 (44.6)	1	-14.8 (9.1)	-24.0 (16.7)
IL	1	-13.3 (13.2)	-27.9 (32.3)	1	-17.2 (30.9)	-34.3 (45.3)	1	-14.6 (8.9)	-23.6 (15.9)
Q learning	1	-13.7 (21.4)	-29.9 (35.1)	1	-11.6 (19.7)	-39.1 (52.9)	1	-14.4 (9.9)	-24.9 (17.5)
reference data	–	-38.8 (39.7)	-45.2 (46.6)	–	-55.3 (51.1)	-59.0 (54.2)	–	-36.8 (24.4)	-36.8 (23.0)

Table 1: Mean and standard deviation (across sequences) of test set forward and reverse rewards for translations. b refers to beam size during inference.

	IWSLT’14 De-En	WMT’16 Ro-En	WMT’14 De-En
p_f (greedy decoding)	33.65 (0.06)	33.23 (0.14)	30.39 (0.13)
p_f (beam search)	34.54 (0.08)	33.98 (0.15)	31.78 (0.08)
BSR	35.43 (0.06)	34.81 (0.09)	32.15 (0.14)
KD	35.39 (0.04)	33.95 (0.10)	31.71 (0.05)
IL	35.61 (0.09)	34.65 (0.07)	31.90 (0.07)
Q learning	34.60 (0.08)	34.31 (0.15)	31.60 (0.19)

Table 2: Test set sacreBLEU (mean & standard deviation of three runs using different random seeds). IL performs the best among the three proposed methods.

	IWSLT’14 De-En	WMT’16 Ro-En	WMT’14 De-En
p_f (greedy decoding)	62.40 (0.04)	61.14 (0.10)	64.83 (0.10)
p_f (beam search)	63.21 (0.07)	61.42 (0.15)	65.79 (0.08)
BSR	64.15 (0.05)	62.67 (0.13)	66.32 (0.12)
KD	63.88 (0.04)	61.78 (0.10)	66.00 (0.07)
IL	63.94 (0.13)	62.35 (0.16)	66.14 (0.08)
Q learning	63.25 (0.07)	61.70 (0.18)	65.92 (0.14)

Table 3: Test set BLEURT-20-D12 (mean & standard deviation of three runs). IL performs the best among the three proposed methods. Significance test is conducted in Table 8, which shows that IL’s scores are significantly better than the scores by beam search; in addition, IL’s scores are not significantly different from BSR’s scores.

and the decoding speed for any of three proposed approaches is also similar.

Rewards. First, comparing the three approaches to greedy decoding or beam search from p_f , we see that the three approaches achieve smaller forward rewards, but much larger reverse rewards. This observation is expected given that the three approaches consider both the forward and reverse rewards, while greedy decoding or beam search from p_f only consider forward rewards. Second, comparing the three approaches against BSR, the three approaches achieve both smaller forward rewards

and smaller reverse rewards. However, we find this a reasonable trade-off to be made between decoding latency and rewards, as all these approaches are 1–2 orders of magnitude faster in decoding.

Among the three approaches, KD and IL achieve a better balance between forward and reverse rewards. This observation can be explained by the difference in how the reverse reward is presented among the three approaches. In KD and IL, the learning signal by reverse rewards is implicitly spread throughout all the steps in a sequence. In other words, changing the conditional distribution in each time-step would adjust the loss in KD and the reverse energies in IL. In Q learning, the reverse reward is sparse: it only appears at the end of the sequence, unlike the forward reward which is spread throughout all the steps. This makes it easier for Q learning to maximize the forward reward compared to the reverse reward which requires many more updates to be propagated toward the earlier time steps.

Translation quality. The three approaches achieve BLEU and BLEURT scores that are comparable to those by BSR. Moreover, the three approaches achieve BLEU scores that are much better than “greedy decoding from p_f ” which has the same computational budget; they are often better than “beam search from p_f ” as well. In particular, Table 3 shows that IL’s BLEURT scores are significantly higher than the scores of beam search across all three datasets. In addition, IL’s BLEURT scores are not significantly different from BSR across all three datasets. Therefore, our approaches are able to generate translations with similar quality as those by BSR, while being 5–7 times as fast as beam search and 50–100 times as fast as BSR.

	IWSLT’ 14 De-En				WMT’ 16 Ro-En				WMT’ 14 De-En			
	b	fwd reward mean (std)	rvs reward mean (std)	BLEU	b	fwd reward mean (std)	rvs reward mean (std)	BLEU	b	fwd reward mean (std)	rvs reward mean (std)	BLEU
$p_{\text{KD}/\text{beam}}$ trained by $(X, \tilde{Y}_{\text{beam}})$	1	-13.3 (13.4)	-31.6 (35.2)	34.80	1	-17.0 (17.4)	-38.9 (49.0)	33.22	1	-14.7 (9.2)	-28.0 (19.3)	31.38
$p_{\text{KD}/\text{nc}}$ trained by $(X, \tilde{Y}_{\text{NC}})$	1	-13.8 (13.9)	-28.0 (32.7)	35.39	1	-17.2 (26.3)	-35.4 (44.6)	33.95	1	-14.8 (9.1)	-24.0 (16.7)	31.71

Table 4: The rewards and BLEU scores using two KD approaches: $p_{\text{KD}/\text{beam}}$ uses the pseudo-corpus generated by doing beam search from p_f . $p_{\text{KD}/\text{nc}}$ uses the pseudo-corpus generated by BSR.

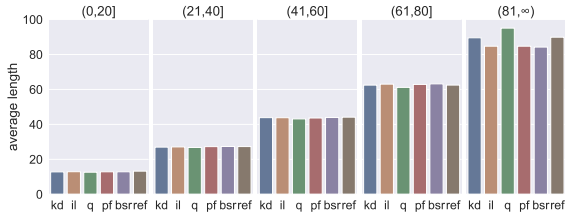


Figure 1: Average length bucketed by length of the source sentence. The five buckets contain 453, 877, 376, 92, 26 sentences, respectively. The six systems are KD, IL, Q learning, beam search by p_f , BSR, and reference translations, respectively. In the longest length bucket, Q learning produces translations that are longer than translations by other systems.

5.2 Analysis of Translations

In Q learning, the reverse reward is only presented as a learning signal at the end of each sequence. As observed earlier by Welleck et al. (2020), the length of the generations may inform us of the possible degeneracies, such as excessive repetitions.

Therefore, we analyze WMT’ 16 Ro-En translations generated by different systems, and we first examine the lengths of translations in different source length buckets. Figure 1 shows that the lengths by different systems are similar in the first four buckets, but in the longest source length bucket $(81, \infty)$, Q learning produces longer translations.

Closer examination of the translations reveal that Q learning produces degenerate translations with extensive repetitions when the source sentences are among the longest in the entire dev set; other models do not have this issue. Some randomly selected examples are shown in Table 6.

To confirm this finding, we analyze repetitions by source-length buckets. We define “token rep” to be the percentage of tokens that have appeared in

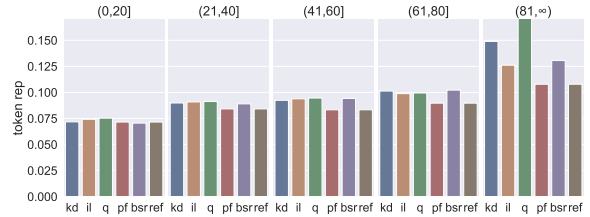


Figure 2: Repetition rate (“token rep”) bucketed by length of the source sentence. The five buckets contain 453, 877, 376, 92, 26 sentences, respectively. N.B.: In the last bucket, “token rep” for Q-generated translations is around 0.31, and the bar is truncated.

the immediately preceding 5-grams:

$$\frac{\sum_{i=1}^N \sum_{t=6}^{T^{(i)}} \mathbb{1} \left[y_t^{(i)} \in \{y_{t-5}^{(i)}, \dots, y_{t-1}^{(i)}\} \right]}{\sum_{i=1}^N \sum_{t=6}^{T^{(i)}} 1},$$

where the superscript indicates the i -th example, and N indicates the number of translations.

We see from Figure 2 that for the longest source-sentence length bucket $(81, \infty)$, Q produces translations with a significantly larger 5-gram repetition rate. Moreover, beam search from the forward only model p_f exhibits a behavior most similar to reference translations. We leave it for the future to study the cause behind an elevated level of repetition in noisy-channel decoding.

system 1	system 2				
	p_f (beam search)	BSR	KD	IL	Q learning
p_f (beam search)	100	–	–	–	–
BSR	81.2	100	–	–	–
KD	64.5	66.0	100	–	–
IL	64.4	66.2	70.8	100	–
Q learning	74.0	72.0	64.3	64.1	100

Table 5: Corpus-level BLEU between translations by pairs of systems. Each reported BLEU is averaged between two directions.

Next, to compare translation similarity among

different approaches, we examine the corpus-level BLEU score between each pair of approaches, averaged between two directions. By Table 5, translations by BSR is similar to those produced by p_f and Q learning, compared to KD and IL. Now we compare the translations produced by the three approaches. Translations by KD are more similar to IL, compared to BSR and Q learning. This is in line with our intuition that KD and IL differ from Q learning, given that how the reverse reward is presented is different between KD/IL and Q learning.

5.3 Further Analysis

KD. One may wonder whether the improvements in KD arise from the KD procedure or because we use BSR when constructing the pseudo-corpus. We therefore experiment with another model $p_{\text{KD}/\text{beam}}$: we generate the pseudo-corpus \tilde{Y}_{beam} from the training set, by beam search from p_f , and then use MLE to train $p_{\text{KD}/\text{beam}}$ using the parallel corpora $(X, \tilde{Y}_{\text{beam}})$. Table 4 suggests that the forward rewards of the two approaches are similar, but the reverse rewards for $p_{\text{KD}/\text{nc}}$ is much larger. Meanwhile, $p_{\text{KD}/\text{nc}}$ produces translations with higher BLEU. It is therefore necessary to use BSR to generate the pseudo-corpus, in order to amortize noisy-channel NMT using KD.

Q learning. Why does Q learning, the best understood approach among the three, fail to achieve rewards that are comparable to BSR? The two challenges of a general deep Q learning algorithm are exploration and optimization.

Exploration refers to whether we can find high-quality trajectories. We hypothesize that it is not an issue given the diversity of trajectories we use, as shown in Appendix A.1. We even attempt adding high-reward trajectories from BSR as well as trajectories from a deep ensemble of multiple p_f 's but neither BLEU nor reward improves.

We thus suspect optimization as a challenge. The reverse reward $\log p_r(\mathbf{x}|\mathbf{y})$ is *sparse* in that it is non-zero only at the terminal state $(\mathbf{y}_{1:T}, \mathbf{x})$ where $y_T = \langle \text{eos} \rangle$. The difficulty in maximizing the sparse reverse reward comes from using one-step bootstrapping in Q learning. Such bootstrapping allows Q learning to cope with very long episodes or even an infinite horizon, but this slows down the propagation of future reward to the past. Because we always work with relatively short episodes only in machine translation, we should investigate other

learning paradigms from reinforcement learning, such as R learning (Mahadevan, 1996). We leave this further investigation to the future.

6 Related Work

One of our approaches adapts knowledge distillation (KD) for the noisy channel NMT setting. KD (Hinton et al., 2015; Kim and Rush, 2016) has been shown to work well for sequence generation. Chen et al. (2018) propose trainable greedy decoding, in which they use knowledge distillation to train a greedy decoder so as to amortize the cost of beam search. More subsequent studies have demonstrated the effectiveness of KD in neural machine translation (Freitag et al., 2017; Tan et al., 2019); Gu et al. (2017) show that it is difficult for on-policy reinforcement learning (RL) to work better than KD. Recently, KD has greatly boosted performance of non-autoregressive MT models (Gu et al., 2018; Lee et al., 2018; Tu et al., 2020). KD is also used to speed up speech synthesis and the approach has been widely deployed in real products (van den Oord et al., 2018).

RL for sequence generation has been greatly inspired by Sutton and Barto (1998). Ranzato et al. (2016) and Bahdanau et al. (2016) apply on-policy RL (REINFORCE and actor-critic algorithms) to MT, but the major optimization challenge lingers given that the reward is usually sparse. Choshen et al. (2020) recently find that the improvements in MT performance may rely on a good initialization. To address the sparsity issue, Norouzi et al. (2016) attempt a hybrid maximum likelihood (ML) and RL approach. More recently, Pang and He (2021) attempt to use an offline RL setting with per-token reward based on the a translator trained using standard MLE.

In recent years, off-policy RL methods have been used to best leverage trajectories in text generation. For instance, in the chatbot setting (Serban et al., 2017; Zhou et al., 2017), the periodically-collected human feedback is treated as the trajectory. In our case, we leverage the expensive BSR-obtained trajectories as well as trajectories from many different models and sources, although the sparse reward issue still lingers.

Finally, we point out a recent endeavor to speed up noisy channel NMT inference (Bhosale et al., 2020). They reduce the size of the channel model, the size of the output vocabulary, and the number of candidates during beam search. Our solution

source: acum , in sa , tsipras cere grecilor sa ii incredinteze din nou mandatul de premier , in cadrul unor alegeri despre care sustine ca ii vor intari pozitia politica .
KD: now , however , tsipras is urging greeks to entrust the prime minister 's mandate again , in an election he claims will strengthen his political position .
IL: now , however , tsipras is asking greeks to reentrust them with the prime minister 's term , in an election that they claim will strengthen his political position .
Q learning: now , however , tsipras is urging greeks to reentrust his term as prime minister in an election that he claims will strengthen his political position .
beam search by p_f : now , however , tsipras is urging greeks to re-entrust the prime minister 's term in an election that he claims will strengthen his political position .
BSR: now , however , tsipras is urging greeks to reentrust the prime minister 's term , in an election that he claims will strengthen his political stance .
reference: now , however , tsipras asks the greeks again to entrust him with the prime minister position , during an election which he says will strengthen his political position .

source: adomnitei a fost trimis in judecata de directia nationala antico<unk> ruptie (dna) , fiind acuzat de favorizarea faptuitorului si fals intelectual dupa ce , spun pro <unk> curorii , ar fi incercat sa mascheze un control de audit in urma caruia se descoperise o serie de nereguli cu privire la receptia dintr-un contract public semnat intre cj si firma laser co .
KD: adomnitei was sued by the national anti-co nistelrooij ruptie (dna) as accused of favouring the perpetrator and false intellectual after , pro nistelrooij curorii says , he would have tried to disguise an audit control as a result of which a number of irregularities concerning reception in a public contract signed between the cj and laser co were discovered .
IL: adomnitei was sued by the national directorate antico iel ruptie (dna) and accused of favouring the perpetrator and forgery an intellectual after , pro iel curorii says , he had tried to disguise an audit control that found a number of irregularities regarding the reception in a public contract signed between cj and laser .
Q learning: the runner the runner , the runner-the runner-in-ranging runner-up is given to the latter , as he is accused of promoting the perpetrator and faltering intellectual after , says pro or: curors , tried to disguise an audit control , as a result of which a number of irregularities concerning a reception signed between cj and laser had been discovered in a public cross-border convoy .
beam search by p_f : adomnitei was sued by the national anti-co nistelrooij ruptie (dna , accused of favouring the perpetrator and forgery intellectual after allegedly attempting to disguise an audit control line between cj and lasco .
BSR: adomnitei was sued by the national anti-co xiated department (dna , accused of favouring the perpetrator and forgery intellectual after allegedly attempting to disguise an audit control line signed between cj and the lasco firm .
reference: adomni<unk> ei was indicted by the national anticorruption directorate (dna) , being accused of favouring the offender and forgery after , according to the prosecutors , he tried to mask an audit which discovered a number of irregularities regarding the acceptance of a public contract entered into by the county council and the company laser co .

Table 6: WMT’16 Ro-En examples produced by different systems. The top example is randomly selected. The bottom example is an example with a long source, and Q learning produces repetitions.

is orthogonal: we aim to use a separate network to amortize decoding cost, while not changing the network’s architecture.

7 Conclusion

We describe three approaches (KD, IL, Q learning) to train an amortized noisy-channel NMT model. We investigate whether greedily decoding from these models will lead to accurate translations in terms of reward and quality. Although all three approaches fail to achieve comparable rewards to BSR, the reverse rewards are much higher than those from non-BSR baselines, often at the expense of forward rewards. However, we found the translation quality (measured by BLEU and BLEURT) to be comparable to that of BSR, while massively speeding up inference. For future work, the research community could further investigate better ways to optimize toward a sparse reward in the language generation context. Another way to approach the Q learning optimization challenge is to find better reward functions including denser rewards.

Acknowledgement

We thank Eneko Agirre, Jon Ander Campos, Kevin Gimpel, Nitish Joshi, Elman Mansimov, and Ethan Perez (alphabetical order) for valuable discussion.

This work was supported by Samsung Advanced Institute of Technology (under the project *Next Generation Deep Learning: From Pattern Recognition to AI*) and NSF Award 1922658 NRT-HDR: FUTURE Foundations, Translation, and Responsibility for Data Science.

References

- Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2016. An actor-critic algorithm for sequence prediction. *arXiv preprint arXiv:1607.07086*.
- Shruti Bhosale, Kyra Yee, Sergey Edunov, and Michael Auli. 2020. [Language models not just for pre-training: Fast online neural noisy channel modeling](#). In *Proceedings of the Fifth Conference on Machine Translation*, pages 584–593, Online. Association for Computational Linguistics.
- Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. [Findings of the 2014 workshop on statistical machine translation](#). In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.
- Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, An-

- tonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. [Findings of the 2016 conference on machine translation](#). In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.
- Eric Brill and Robert C. Moore. 2000. [An improved error model for noisy channel spelling correction](#). In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 286–293, Hong Kong. Association for Computational Linguistics.
- Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2014. Report on the 11th IWSLT evaluation campaign. In *Proceedings of the International Workshop on Spoken Language Translation*, volume 57, Hanoi, Vietnam.
- Peng-Jen Chen, Ann Lee, Changhan Wang, Naman Goyal, Angela Fan, Mary Williamson, and Jiatao Gu. 2020. [Facebook AI’s WMT20 news translation task submission](#). In *Proceedings of the Fifth Conference on Machine Translation*, pages 113–125, Online. Association for Computational Linguistics.
- Yun Chen, Victor O.K. Li, Kyunghyun Cho, and Samuel Bowman. 2018. [A stable and effective learning strategy for trainable greedy decoding](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 380–390, Brussels, Belgium. Association for Computational Linguistics.
- Leshem Choshen, Lior Fox, Zohar Aizenbud, and Omri Abend. 2020. [On the weaknesses of reinforcement learning for neural machine translation](#). In *International Conference on Learning Representations*.
- Abdessaamad Echihabi and Daniel Marcu. 2003. [A noisy-channel approach to question answering](#). In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 16–23, Sapporo, Japan. Association for Computational Linguistics.
- Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *arXiv preprint arXiv:1702.01802*.
- Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. [Non-autoregressive neural machine translation](#). In *International Conference on Learning Representations*.
- Jiatao Gu, Kyunghyun Cho, and Victor O.K. Li. 2017. [Trainable greedy decoding for neural machine translation](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1968–1978, Copenhagen, Denmark. Association for Computational Linguistics.
- Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. [Distilling the knowledge in a neural network](#). In *NIPS Deep Learning and Representation Learning Workshop*.
- Frederick Jelinek. 1997. *Statistical methods for speech recognition*. MIT press.
- Yoon Kim and Alexander M. Rush. 2016. [Sequence-level knowledge distillation](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Tom Kocmi, Tomasz Limisiewicz, and Gabriel Stanovsky. 2020. [Gender coreference and bias evaluation at WMT 2020](#). In *Proceedings of the Fifth Conference on Machine Translation*, pages 357–364, Online. Association for Computational Linguistics.
- Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. [Statistical phrase-based translation](#). In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133.
- Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. [Deterministic non-autoregressive neural sequence modeling by iterative refinement](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Brussels, Belgium. Association for Computational Linguistics.
- Sridhar Mahadevan. 1996. Average reward reinforcement learning: Foundations, algorithms, and empirical results. *Machine learning*, 22(1):159–195.
- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.
- Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. [Facebook FAIR’s WMT19 news translation task submission](#). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.
- Mohammad Norouzi, Samy Bengio, Zhifeng Chen, Navdeep Jaitly, Mike Schuster, Yonghui Wu, and Dale Schuurmans. 2016. [Reward augmented maximum likelihood for neural structured prediction](#). In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

- Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. [fairseq: A fast, extensible toolkit for sequence modeling](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.
- Richard Yuanzhe Pang and He He. 2021. [Text generation by learning from demonstrations](#). In *International Conference on Learning Representations*.
- Matt Post. 2018. [A call for clarity in reporting BLEU scores](#). In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Marc’Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations*.
- Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. [BLEURT: Learning robust metrics for text generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.
- Iulian V Serban, Chinnadhurai Sankar, Mathieu Germain, Saizheng Zhang, Zhouhan Lin, Sandeep Subramanian, Taesup Kim, Michael Pieper, Sarath Chandar, Nan Rosemary Ke, et al. 2017. A deep reinforcement learning chatbot. *arXiv preprint arXiv:1709.02349*.
- Richard S Sutton. 1990. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine learning proceedings 1990*, pages 216–224. Elsevier.
- Richard S Sutton and Andrew G Barto. 1998. *Reinforcement learning: An introduction*. MIT press.
- Xu Tan, Yi Ren, Di He, Tao Qin, and Tie-Yan Liu. 2019. [Multilingual neural machine translation with knowledge distillation](#). In *International Conference on Learning Representations*.
- Chau Tran, Shruti Bhosale, James Cross, Philipp Koehn, Sergey Edunov, and Angela Fan. 2021. [Facebook AI’s WMT21 news translation task submission](#). In *Proceedings of the Sixth Conference on Machine Translation*, pages 205–215, Online. Association for Computational Linguistics.
- Lifu Tu, Richard Yuanzhe Pang, Sam Wiseman, and Kevin Gimpel. 2020. [ENGINE: Energy-based inference networks for non-autoregressive machine translation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2819–2826, Online. Association for Computational Linguistics.
- Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis Cobo, Florian Stimberg, Norman Casagrande, Dominik Grewe, Seb Noury, Sander Dieleman, Erich Elsen, Nal Kalchbrenner, Heiga Zen, Alex Graves, Helen King, Tom Walters, Dan Belov, and Demis Hassabis. 2018. [Parallel WaveNet: Fast high-fidelity speech synthesis](#). In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3918–3926. PMLR.
- Christopher JCH Watkins and Peter Dayan. 1992. Q-learning. *Machine learning*, 8(3-4):279–292.
- Sean Welleck, Ilya Kulikov, Jaedeok Kim, Richard Yuanzhe Pang, and Kyunghyun Cho. 2020. [Consistency of a recurrent language model with respect to incomplete decoding](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5553–5568, Online. Association for Computational Linguistics.
- Kyra Yee, Yann Dauphin, and Michael Auli. 2019. [Simple and effective noisy channel modeling for neural machine translation](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5696–5701, Hong Kong, China. Association for Computational Linguistics.
- Lei Yu, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Tomas Kocisky. 2017. [The neural noisy channel](#). In *International Conference on Learning Representations*.
- Lei Yu, Laurent Sartran, Po-Sen Huang, Wojciech Stokowiec, Domenic Donato, Srivatsan Srinivasan, Alek Andreev, Wang Ling, Sona Mokra, Agustin Dal Lago, Yotam Doron, Susannah Young, Phil Blunsom, and Chris Dyer. 2020. [The DeepMind Chinese–English document translation system at WMT2020](#). In *Proceedings of the Fifth Conference on Machine Translation*, pages 326–337, Online. Association for Computational Linguistics.
- Li Zhou, Kevin Small, Oleg Rokhlenko, and Charles Elkan. 2017. End-to-end offline goal-oriented dialog policy learning via policy gradient. *arXiv preprint arXiv:1712.02838*.

A More Information on Q learning for Amortized Noisy Channel NMT

A.1 Details on trajectories

We have obtained trajectories from different sources in the off-policy algorithm (Algorithm 1). Each trajectory contains a source-language sequence \mathbf{x} , a target-language sequence \mathbf{y} , and the corresponding sequence of rewards $\mathbf{r} = (r_1, \dots, r_T)$.

One natural category of trajectories to consider is the ones obtained by Q during training. Source (1a) and source (1b) correspond to Q -based trajectories.

Source (2) corresponds to p_f -obtained trajectories. Specifically, we split this category into a few sub-sources. (2a) The \mathbf{y} is obtained through sampling from p_f with temperature sampled from $\text{Uniform}([0, 1])$. (2b) The \mathbf{y} is obtained through greedily decoding from p_f . (2c) The \mathbf{y} is obtained through beam search from p_f with a beam size randomly chosen from 2 to 10. (2d) The \mathbf{y} is obtained through beam search from p_f : we first obtain 50 candidate sequences corresponding to largest p_f probabilities using beam search with beam size 50; next, we pick a random sequence out of these 50 sentences.

We have also experimented with gold-standard trajectories from the parallel translation dataset \mathcal{D} , but the inclusion of such trajectories do not lead to better rewards (of translations generated from Q).

The probability for using (1a), (1b), (2a), (2b), (2c), (2d) sequences are 0.3, 0.2, 0.2, 0.1, 0.1, 0.1, respectively.

B More Discussion on Experiments

BSR hyperparameters. γ is tuned in $\{0.1, 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5\}$, and b is tuned in $\{5, 10, 20, \dots, 100\}$ for the first two datasets and $\{5, 10, 20, \dots, 50\}$ for WMT’14 De-En due to memory constraints. The best γ is 0.9, 0.5, 0.5, for IWSLT’14 De-En, WMT’16 Ro-En, WMT’14 De-En, respectively; the best b is 100, 70, 50 for the three datasets, respectively.

Details on p_f and p_r . Recall that p_f is the forward translator (from the source language to the target language) and p_r is the reverse translator (from the target language to the source language). We use transformer-based architectures for all experiments. Refer to Table 7 for the architecture.

Number of parameters in the models. The IWSLT’14 De-En transformer has 39,469,056 parameters, the WMT’16 Ro-En transformer has 62,046,208 parameters, and the WMT’14 De-En transformer has 209,911,808 parameters.

Discussion on Q learning. In Section 5.3, to investigate whether better trajectories can improve Q learning results, we attempt adding high-reward trajectories from BSR as well as trajectories from a deep ensemble of two p_f ’s. Deep ensembling two models (using different seeds) can produce high-quality translations. In this case, we simply want to use deep ensembling to diversify the sources of high-reward and high-BLEU trajectories. However, the result is that neither BLEU nor reward improves.

C Ethical Considerations

IWSLT and WMT datasets are standard machine translation benchmarks. The datasets come from a variety of sources: phone conversations, parliament proceedings, news, and so on. There may be naturally occurring social biases in the datasets which have not undergone thorough cleansing. Training on these potential biases may lead to biased generations. There has been recent work studying such biases (Kocmi et al., 2020).

The standard practice of creating the pseudo-corpus requires a significant amount of computation. This step is optional, but it gives a boost in performance. We argue that if the MT system is put into production, then the benefit from the efficient inference will outweigh the cost of generating the pseudo-corpus.

	IWSLT' 14 De-En	WMT' 16 Ro-En	WMT' 14 De-En
encoder embedding dimension	512	512	1,024
number of encoder attention heads	4	8	16
encoder ffn embedding dimension	1,024	2,048	4,096
encoder layers	6	6	8
decoder embedding dimension	512	512	1,024
number of decoder attention heads	4	8	16
decoder ffn embedding dimension	1,024	2,048	4,096
decoder layers	6	6	8
learning rate	0.0005	0.0005	0.0005
dropout rate	0.3	0.1	0.1
# tokens in a batch	4,096 (2^{12})	65,536 (2^{16})	65,536 (2^{16})

Table 7: Settings for the forward model p_f and the reverse (channel) model p_r .

	IWSLT' 14 De-En	WMT' 16 Ro-En	WMT' 14 De-En
p_f (greedy)	62.40 (0.04)	61.14 (0.10)	64.83 (0.10)
p_f (beam)	63.21 (0.07)	61.42 (0.15)	65.79 (0.08)
BSR	64.15 (0.05)	62.67 (0.13)	66.32 (0.12)
KD	63.88 (0.04) *	61.78 (0.10) *	66.00 (0.07) *
IL	63.94 (0.13) *†	62.35 (0.16) *†	66.14 (0.08) *†
Q learning	63.25 (0.07)	61.70 (0.18)	65.92 (0.14)

Table 8: Test set BLEURT-20-D12 (mean & standard deviation of three runs). IL performs the best among the three proposed methods. *: The score is significant (p-value smaller than 0.05) compared to the beam search results. †: The score is significantly higher (p-value smaller than 0.05) than BSR results, or the score is not significantly different (p-value larger than 0.05) from the BSR results.

Math Word Problem Generation with Multilingual Language Models

Kashyapa Niyarepola, Dineth Athapaththu, Savindu Ekanayake, Surangika Ranathunga

Department of Computer Science and Engineering, University of Moratuwa

Katubedda 10400, Sri Lanka

[kashyapabandara.17, dinethnaradaam.17,
savinduekanayake.17, surangika]@cse.mrt.ac.lk

Abstract

Auto regressive text generation for low-resource languages, particularly the option of using pre-trained language models, is a relatively under-explored problem. In this paper, we model Math Word Problem (MWP) generation as an auto-regressive text generation problem. We evaluate the pre-trained sequence-to-sequence language models (mBART and mT5) in the context of two low-resource languages, Sinhala and Tamil, as well as English. For the evaluation, we create a multi-way parallel MWP dataset for the considered languages. Our empirical evaluation analyses how the performance of the pre-trained models is affected by the (1) amount of language data used during pre-training, (2) amount of data used in fine-tuning, (3) input seed length and (4) context differences in MWPs. Our results reveal that the considered pre-trained models are capable of generating meaningful MWPs even for the languages under-represented in these models, even though the amount of fine-tuning data and seed length are small. Our human evaluation shows that a Mathematics tutor can edit a generation question fairly easily, thus highlighting the practical utility of automatically generating MWPs.

1 Introduction

Despite being one of the most important subjects, many school children find Mathematics difficult (Acharya, 2017), with many exams reporting high failure rates in Mathematics (Rylands and Coady, 2009). One way of improving Mathematics skills is to practice solving Mathematics problems (Thompson, 1985). However, this places extra burden on the tutors - they have to create different Mathematics questions and grade student answers. The alternative is to automatically generate Mathematics questions and grade student answers. The need of such systems that support as many languages as possible, is even more pronounced during the times of pandemics and war, where students

get confined to homes/shelters without access to physical schools.

In this paper, we focus on the problem of automatically generating Mathematical Word problems (MWPs). Considering the fact that learning Mathematics is not a privilege to students speaking a particular language, we want to investigate the possibility of MWP generation in multiple languages. An MWP is a “narrative with a specific topic that provides clues to the correct equation with numerical quantities and variables therein” (Zhou and Huang, 2019). MWPs can be in categories such as algebra, geometry and statistics. An elementary MWP written in English is shown in the below example.

Rosy made cookies and she used 2 kg flour and 1.5 kg sugar. How much more flour than sugar did Rosy use?

Early solutions to MWP generation relied on template-based approaches (Polozov et al., 2015), and question rewriting (Koncel-Kedziorski et al., 2016). More recently, Recurrent Neural Networks (RNN) (Zhou and Huang, 2019; Liyanage and Ranathunga, 2020), fine-tuning pre-trained language models (Wang et al., 2021) as well as Variational Autoencoders (VAE) (Liu et al., 2020; Cao et al., 2021) have been proposed. Only Liyanage and Ranathunga (2020) have tried their NN solution in a multilingual setting, however the results are sub-optimal.

Thus, our objective is to investigate the use of multilingual pre-trained models for MWP generation. Here, we treat MWP generation as an auto-regressive problem - the system has to generate a question starting with the provided seed (the starting portion of the question that is expected to be generated). Compared to text generation tasks such as story generation (Roemmele, 2016) or news generation (Leppänen et al., 2017), MWP generation is challenging because MWPs have mathematical constraints, units and numerical values as shown in

the above example.

As mentioned above, auto-regressive language models such as GPT-x (Radford et al., 2019) have been already used for MWP generation (Wang et al., 2021). They are a common choice for Natural Language Generation (NLG) tasks (Lee and Hsiang, 2020; Mosallanezhad et al., 2020; Budzianowski and Vulić, 2019). Sequence-to-sequence models such as BART (Lewis et al., 2019) and T5 (Raffel et al., 2019) have also been used for NLG in an auto-regressive manner (Tan et al., 2020; Lewis et al., 2020). However, this option has been used to a lesser extent compared to GPT-x in similar text generation tasks, and never for MWP generation.

Despite their success on English text generation, GPT-x models are not available for other languages. Building multilingual or language-specific GPT models is not practical for many languages, particularly the low-resource ones. In contrast, T5 and BART both have their multilingual versions: mT5 (Xue et al., 2020) and mBART (Tang et al., 2020) (respectively). We are only aware of the empirical analysis of Chen et al. (2021), who tested the auto-regressive text generation capabilities of mT5 and mBART in the context of 4 high resource languages (for four tasks: story, question and title generation).

We carry out an empirical study on the mBART and mT5 models for MWP generation, considering two low-resource languages Sinhala and Tamil, along with English. All these languages are included in mBART and mT5. For a more comprehensive analysis, we evaluate T5, BART and GPT-2 for English MWP generation as well. Our experiments answer four important questions:

1. How the performance of mT5 and mBART varies depending on the language - because, for the related Machine Translation task, it has been shown that the model performance on individual languages depends on the amount of language-specific data used during model pre-training (Lee et al., 2022)
2. How the performance of the models varies depending on the amount of fine-tuning data - because for many languages, having a large training set is not realistic
3. How much information (size of the seed) should be provided to the model at the inference stage for it to generate a meaningful

MWP - because a tutor should be able to generate a new MWP by providing minimal information.

4. How the context of an MWP affects the generation performance - because there is a wide variety of MWPs

As an additional contribution, we create a benchmark dataset by extending the dataset created by Liyanage and Ranathunga (2020) for MWP generation. Each English question was manually translated to Sinhala and Tamil, creating a multi-way parallel dataset. Our dataset is publicly released¹, and can be considered as a test set even for Machine Translation.

We believe that our work is the first to conduct an empirical analysis on the use of (1) GPT, BART, T5, mBART and mT5 for auto-regressive generation of MWPs and (2) mBART and mT5 for the general task of auto-regressive text generation considering low-resource languages. Our findings are indeed very promising for low-resource languages. Even for very small seeds and fine-tuning dataset sizes, these models (mBART in particular) yield very good results with very little grammar and spelling errors. Thus we can present the use of these models as a very promising avenue for auto-regressive text generation for low-resource languages, at least for those that are included in the pre-trained models.

2 Related Work

2.1 MWP Generation

Previous research has addressed the problem of MWP generation using three main techniques: question rewriting, template-based generation and text generation with Neural Networks (NNs).

Question rewriting technique rewrites a human-written question by replacing words with new ones from different contexts (Koncel-Kedziorski et al., 2016). However, the numerical values in all the rewritten questions are the same.

In the template-based techniques, first a question template is either provided by a tutor (Nandhini and Balasundaram, 2011; Polozov et al., 2015; Wang and Su, 2016), or generated from an MWP (Bekele, 2020). Most of these template-based techniques are long and tedious processes, with some requiring language specific tools or resources.

¹https://huggingface.co/datasets/NLPC-UOM/MWP_Dataset

Zhou and Huang (2019) present a Deep Neural Network model that has two encoders and one decoder, all based on RNNs. The equation encoder takes in an equation template, and the topic encoder takes in a topic (context). The system is trained in a supervised manner, using an MWP dataset. Thus, for training purposes, the equation and the topic of each training MWP has to be extracted. Wang et al. (2021) also take in an equation and context, however MWP generation is done using GPT-2. Additionally, they introduce constraints to satisfy equation and context correctness. Liu et al. (2020) also take in an equation as the input. However, they expect an external knowledge graph to represent the context. Both the knowledge graph and the equation are encoded using a Convolutional Gated Neural Network model. A Variational Auto-Encoder (VAE) is used to generate the MWP from this encoding. Cao et al. (2021) also make use of a VAE to bridge the gap between abstract math tokens and text. In addition to the equation and common sense knowledge graph as input, they take in the question text, as well as a set of words representing a topic.

In contrast to above research, Liyanage and Ranathunga (2019, 2020) train a single RNN encoder in an auto-regressive manner using the MWP text. Liyanage and Ranathunga (2019) impose Mathematical constraints during post processing, while Liyanage and Ranathunga (2020) achieve the same using POS embeddings as input to the model. As for NN-based solutions, only Liyanage and Ranathunga (2019, 2020) considered MWP generation in languages other than English.

2.2 Bench-marking NLG with Pre-trained Models

NLG is an umbrella term used for a set of tasks where the objective is to generate a text as the output. In addition to auto regressive text generation, NLG covers tasks such as text summarization, text simplification, and graph to text generation. The GEM benchmark (Gehrmann et al., 2021) evaluates BART, T5, mBART and mT5 for 11 different NLG tasks. However, there is no evaluation on an auto regressive text generation task. Moreover, except for one dataset, all the others are focused only on high-resource languages. The GLGE benchmark (Liu et al., 2021), which evaluated BART and MASS pre-trained models also does not have a dataset for auto regressive text generation. Further,

evaluation is only done for English.

Several shared tasks have been organized for multilingual NLG tasks such as surface realization (Mille et al., 2020) and RDF triples to text (Ferreira et al., 2020). Submissions to these shared tasks have experimented with various pre-trained models. However, the datasets focus only on high and mid-resource languages. In contrast to the above datasets, Kumar et al. (2022)’s multilingual NLG dataset suit covers many low-resource Indic languages. They use mT5 and IndicBART for evaluation. However, an auto regressive text generation task is not included in this suit. As for auto-regressive text generation evaluation, we are only aware of Chen et al. (2021), who considered mT5 and mBART. However, evaluation was done only on 4 high-resource languages.

3 Methodology

All the models considered in this research are trained using the Transformer architecture (Vaswani et al., 2017), which is an Encoder-Decoder model that contains a set of encoder layers and decoder layers. GPT, BART and T5 are pre-trained with English data. mBART and mT5 are pre-trained with data from multiple languages (50 and 101, respectively). Here, pre-training means, the models have been trained with a self-supervised objective such as ‘span corruption’ (Xue et al., 2020). All these models have to be fine-tuned for the selected downstream task.

GPT models are decoder based. Here, the encoder-decoder cross attention block is discarded because there is no encoder. Self-attention has been replaced by masked self-attention. We follow the standard training procedure of GPT-2 model in training it for MWP generation. T5, BART, mBART and mT5 are encode-decoder models. They expect a text sequence as the input and output. For auto-regressive text generation, we use the conditional generator option of BART/mBART and T5/mT5, which makes the output of the model conditioned on the preceding input sequence. In both these cases, the models generate the rest of the MWP for a given seed.

4 Experiments

4.1 Dataset

Liyanage and Ranathunga (2020)’s dataset contains two types of MWPs: simple MWPs and algebraic MWPs. The simple MWP dataset contains 2000

questions and the Algebraic MWP dataset contains 2350 questions. This dataset contains questions in English, Tamil and Sinhala, but is not multi-way parallel.

We extended this dataset using the Dolphin18K dataset (Huang et al., 2016) and the allArith dataset (Roy and Roth, 2016) to add more diversity to the dataset. We selected questions that are similar or slightly higher in complexity compared to Liyanage and Ranathunga (2020)’s corpus. Questions that have lengthy descriptions and those corresponding to complex Mathematical equations were omitted. The extended dataset now contains 4210 Algebraic MWPs and 3160 simple MWPs. Simple MWP dataset contains simple arithmetic questions as the example shown in the introduction. These questions contain constraints such as ‘*first number is always larger than the second one*’. Algebraic MWPs are more logical and require two or more equations to solve.

E.g.: *The sum of two numbers is twenty-three, and the larger number is five more than the smaller number. Find these numbers.*

Corresponding Sinhala and Tamil examples are given in the Figure 1 in Appendix.

Table 1: Statistics of the multi-way parallel dataset

Dataset type	Avg. Num. of words per question	Avg. Num. of characters per question
English Simple (ES)	15	54
English Algebraic (EA)	14	62
Sinhala Simple (SS)	19	61
Sinhala Algebraic (SA)	17	59
Tamil Simple (TS)	13	49
Tamil Algebraic (TA)	16	57

Mathematics tutors translated these questions to Sinhala and Tamil. They were asked to retain the same sentence count and syntactic structure as the English source question, as much as possible. On average, there are two sentences per question, with a maximum of four sentences. Other statistics of the dataset are given in Table 1.

In order to verify the quality of the manual translations, we used the Direct Assessment (DS) method (Bojar et al., 2016). We selected three bilingual speakers (undergraduate students who are proficient in Mathematics) for each language pair (English-Sinhala, English-Tamil). Each evaluator was assigned 200 translated MWPs along with the original English question. They were asked to rate the translated version with respect to adequacy and

Table 2: Quality estimation results of the translated dataset

Data set	Rank					
	0-10	11-29	30-50	51-69	70-90	91-100
SS	0%	1.6%	3%	6.3%	22.6%	66%
SA	0%	0%	0.3%	2.6%	12.6%	84.3%
TS	0%	1%	4%	8.3%	27.6%	59%
TA	7%	12%	6.3%	6%	11.3%	57%

Table 3: Language Coverage of pre-trained models

Model		English	Tamil	Sinhala
BART	Storage(GB)	160	-	-
T5	Storage(GB)	700	-	-
mT5	Token(B)	2733	3.4	0.8
	Pages(M)	3,067	3.5	0.5
mBART	Token(B)	55.61	0.595	0.243
	Storage(GiB)	300.8	12.2	3.6

fluency and give a rating between 1-100, where 0-10: incorrect translation, 11-29: a translation with few correct keywords, but the overall meaning is different from the source, 30-50: a translation with major mistakes, 51-69: a translation which is understandable and conveys the overall meaning of the source but contains typos or grammatical errors, 70-90: a translation that closely preserves the semantics of the source sentence and 91-100: a perfect translation (Bojar et al., 2016). As shown in Table 2, except for the Tamil Algebraic dataset, all the others report a quality level greater than 85.

4.2 Model Selection

According to Huggingface², GPT2-Medium, T5-base and BART-large variants have approximately 300M model parameters. Therefore these were used for further experiments. For multilingual MWP generation, we selected mT5-base and mBART50-large models, to correspond to their monolingual counterparts. As shown in Table 3, Sinhala and Tamil are largely under-represented in both multilingual models.

4.3 Experiment Setup

Fine-tuning for the selected Huggingface models was set-up with 20 epochs, 4-batch size and 1e-4 learning rate. All the experiments were done on a system that has 15 Intel(R) Core(TM) i9-9900K CPUs and Quadro RTX 6000 GPU with 24GB memory.

²https://huggingface.co/transformers/v3.3.1/pretrained_models.html

4.4 Evaluation Metrics

Test BLEU (Papineni et al., 2002) and ROUGE (ROUGE-1 and ROUGE-2) (Lin, 2004) scores were used as the automatic evaluation metrics, as they are still very commonly used (Gehrmann et al., 2021). For all the experiments, we use BLEU-1 for results analysis, with ROUGE results reported in the Appendix. We note that results reported via these two metrics show a correlation.

The generated MWP’s should have correct spelling/grammar and satisfy different Mathematical constraints. A Maths tutor should be able to edit a generated MWP in less time compared to writing a question from scratch. We carried out a human evaluation to validate the quality of the generated questions and the time taken by a tutor to correct a generated MWP.

5 Results and Evaluation

5.1 Pre-trained models vs Baseline

Since Liyanage and Ranathunga (2020) have provided the evaluation results for their dataset of English, Tamil and Sinhala, we considered this as our baseline. Our first experiment is to determine whether fine-tuning the pre-trained models is better than the selected RNN baseline.

For this experiment, we used only Liyanage and Ranathunga (2020)’s dataset, and used the same data split (train:validation:test 80:10:10) they have used³. Note that for English, results are obtained using the monolingual models.

As mentioned earlier, during training and inference of auto-regressive text generation models, the input to the model is the initial portion of text. This is called a *seed*. In this experiment, we tested our models with a quarter of a question (quarter seed). In contrast, Liyanage and Ranathunga (2020) used the first (50-100) characters. Usually, this attributed to more than half of the question. Note that this means the length of the seed varies from question to question.

Results are shown in Table 4. All our models, even when using just the quarter seed, outperform the baseline by a significant margin, thus highlighting the robustness of the pre-trained models even for low-resource language text generation. Sample questions generated from the models are shown in Table 5. Here, compared to the output of the pre-trained models, the question generated by the

baseline is incomplete, not in a question format and has spelling errors.

Table 4: BLEU for the baseline experiments of English, Sinhala and Tamil MWP’s.

Dataset type	Model	Seed size	En	Si	Ta
Simple	Baseline	>Half	22.97	24.49	20.74
	GPT-2	Quarter	67.00	-	-
	BART/mBART	Quarter	80.93	74.52	71.07
	T5/mT5	Quarter	88.42	68.02	66.45
Algebraic	Baseline	>Half	33.53	-	-
	GPT-2	Quarter	48.93	-	-
	BART/mBART	Quarter	62.99	58.13	68.21
	T5/mT5	Quarter	72.69	47.19	55.33

5.2 Effect of Fine-tuning Dataset Size

We conducted comprehensive experiments on our models to analyze how the quality of the results varies with different fine-tuning dataset sizes. We split the dataset for train:validate:test in such a manner that the training set has 80, 40, and 20 percent of the total dataset per MWP category, and conducted three experiments. Validation and test sets were always kept to be 10% of the total dataset per MWP category. Results are shown in Table 6.

The obvious observation is that the performance of all the models drop when the fine-tuning dataset size drops, which of course is not surprising.

As for English auto-regressive text generation results with monolingual models, both sequence-to-sequence models outperform GPT-2. This is in line with observations for other types of text generation tasks such as graph-to-text generation and question answering (Ribeiro et al., 2021). Further, T5 outperforms BART. We believe this is due to T5 being trained with more data, and this observation confirms with what has been reported for tasks such as machine reading comprehension (Tanaka et al., 2021) and text summarization (Garg et al., 2020). English results with mBART and mT5 lag behind their monolingual counterparts. This is to be expected - the multilingual models do not have English data in the same quantities as their monolingual counterparts. However, this lag is usually around 2 BLEU.

As for multilingual models, mBART outperforms mT5 in all the cases except for the 20% train set scenario of the English Algebraic dataset.

³They reported results only using BLEU

Table 5: Sample English MWP’s generated using the baseline and the fine-tuned models. Seed size: Quarter of the question

Model	Generated MWP’s
Reference	The sum of two numbers is 56, their difference is 22, Find the larger number.
Baseline	the sum of two numbers is 12. their different are the two consecutive integers if the sum of the second integers is 10.
Fine-tuned GPT2	The sum of two numbers is 76, the second is 8 more than 3 times first, what are these 2 numbers?
Fine-tuned BART	The sum of two numbers is 60. three times the smaller number minus twice the larger number is 56. Find the larger number.
Fine-tuned T5	The sum of two numbers is 91. the larger number is 1 more than 4 times the smaller number. Find the numbers.

Table 6: Effect of the fine-tuning Dataset Size reported in BLEU (for quarter seed length)

Dataset size	Train Size	Test Size	English					Tamil		Sinhala	
			GPT2	BART	T5	mBART	mT5	mBART	mT5	mBART	mT5
ALG 4210	3370 (80%)	420 (10%)	55.88	60.22	65.32	67.06	62.78	52.68	50.65	45.46	42.44
	1679 (40%)	420 (10%)	54.23	57.76	62.2	60.76	58.86	50.344	49.34	42.58	38.32
	835 (20%)	420 (10%)	51.87	54.93	59.64	53.27	56.34	47.37	42.26	41.03	34.26
SIM 3160	2530 (80%)	316 (10%)	57.65	65.13	67.82	67.74	66.67	65.85	61.67	65.44	61.71
	1264 (40%)	316 (10%)	55.56	57.99	64.43	64.08	62.25	60.24	58.60	60.48	54.08
	632 (20%)	316 (10%)	54.48	55.52	62.09	61.47	57.13	59.5	53.87	56.81	50.92

This is surprising, because as reported in Table 3, mT5 has more Sinhala and Tamil data compared to mBART. Noting that mT5 has more language coverage than mBART, one possible reason for this could be the problem of *curse of multilinguality* - where the cross-lingual transfer in a multilingual model degrades when the language coverage increases in a model (Conneau et al., 2019).

5.3 Effect of Pre-training Dataset Size

An interesting observation is that, although the dataset is multi-way parallel, the result of a model for the same train-test split is not the same across languages. This difference is the highest for the algebraic dataset. Specifically, always English has the highest result, followed by Tamil, and then Sinhala. We attribute this to the amount of language data included in model pre-training (refer Table 3). Moreover, the results gap between Sinhala and English is higher for mT5 compared to mBART. This could be due to the effect of curse of multilinguality that we mentioned earlier - sufficient cross-lingual transfer does not happen between Sinhala and English due to mT5’s high language coverage.

5.4 Effect of the Context of MWP’s

We note that all the models find the algebraic MWP generation more difficult than simple MWP generation. This indicates that text generation capabilities of pre-trained models depend on the context of the text - algebraic MWP’s have more Mathematical context than the simple MWP’s, which contain more open-domain text that is similar to the text used to pre-train the models.

This may be the reason for the simple MWP dataset to have less language-wise difference in model performance compared to the Algebraic dataset as discussed above - the maximum difference is about 5 BLEU between the best performing English and least performing Sinhala. Given the context of simple MWP’s is more similar to the pre-training data, simple MWP generation benefits better from cross-lingual knowledge transfer between related languages.

In order to further evaluate this effect, we carried out an additional experiment - for the 40%-50% train-test split, we trained the models with one dataset, and tested with the other. Results are reported in Table 7. Compared to the results re-

ported in Table 6, we see a substantial drop in the results, when the models are fine-tuned with the other dataset. This highlights the model’s inability to generalize to the general problem of MWP generation, if the dataset contains MWPs only representing a specific context.

Table 7: BLEU score results for different domain train and test sizes

Train ID	Train Size	Test ID	Test Size	mBART	mT5
SA	1679 (40%)	SS	1580 (50%)	32.39	29.23
SS	1264 (40%)	SA	2088 (50%)	27.01	17.87
TA	1679 (40%)	TS	1580 (50%)	35.27	33.44
TS	1264 (40%)	TA	2088 (50%)	32.12	27.75

5.5 Zero-shot MWP Generation

Motivated by the results we obtained in Table 6 for small amounts of fine-tuning data, we carried out zero-shot text generation experiments. However, as seen in Table 8, all the models miserably fail on zero-shot text generation. The sample generations shown in Table 9 evidence that the generated sentences are not questions but more like stories. This is because these pre-trained models are not specifically trained on a question-type dataset. However, when fine-tuned with just 100 data samples, the performance increases by a significant margin. This result agrees with the observations of Burnyshev et al. (2021) on few-shot text generation of task-oriented utterances. This provides a ray of hope for low-resource languages - at least for those that are covered by pre-trained multilingual models, even with a very small training dataset, a descent result can be expected. We also note that this zero-shot/few-shot observation is in-line with those reported for other pre-trained models such as mBERT (Lauscher et al., 2020).

5.6 Effect of Seed Length

The next experiment is to determine the impact of seed length. For this, we fixed the train set size to 40% and tested with 50% of the dataset. Experiments are run on mBART, which is shown to outperform mT5. We varied the seed length from 10%-40%. Table 10 reports the results for Sinhala and Tamil. As expected, the quality of the generated text goes up when the seed length increases. However, even 10% of the seed is enough to pro-

Table 8: Zero-shot and few-shot results for Sinhala and Tamil

Test Dataset	Train Size	Test Size	mBART	mT5
ES	0	986	5.96	0.05
EA	0	1175	8.50	0.42
SS	0	986	6.37	0.01
SA	0	1175	7.50	0.03
TS	0	986	4.57	0.02
TA	0	1175	6.54	0.03
ES	100	986	23.24	4.30
EA	100	1175	34.50	3.93
SS	100	986	52.72	5.42
SA	100	1175	18.21	2.36
TS	100	986	48.86	2.87
TA	100	1175	39.95	0.60

Table 9: Sample Zero shot Generation results

Model	Generated MWPs
Reference	The difference between two numbers is 24, Find the numbers if their sum is 88.
GPT2	The difference between a "first," and an ordinary, job is that the former often requires significant skills.What’s next?Well... not much really right nowthough!
BART	The... The difference between the two
T5	The difference between the two is that the difference between the two is the difference between the

vide an acceptable result - the lowest is 30 BLEU reported for Sinhala Algebra MPW dataset. The impact of question type and the pre-training data amount of the language can be seen here as well.

Table 10: Text generation results for different seed sizes

Seed size	SS	TS	SA	TA
10%	48.9	45.48	30.19	36.77
20%	58.25	57.74	39.91	45.82
30%	65.47	65.02	47.38	54.21
40%	71.51	72.39	53.85	62.5

5.7 Human Evaluation

We analysed the questions generated by the different models to identify the types of errors in MWP generation. The identified errors are given in Table 11.

We also wanted to identify the actual utility of

Table 11: Identified errors in the generated MWPs

Error Type	Description	Example
Co-reference	inconsistent co-reference	<i>Murali had 9 balls in his house and his friend gave him 4. How many balls does Sam have?.</i> Here, the second sentence has the proper noun Sam, instead of Murali
Unit	A numerical quantity is associated with an inconsistent unit	<i>Kamal built a house and he used 90 kg cement and 40 l sand. How much more cement than sand did Kamal use?.</i> Here, sand is given the unit liter (l), instead of kg
Spelling	Spelling mistakes in a word	<i>What three consecutie odd integers have a sum of -105?</i> Word 'consecutie' is misspelled.
Grammar	A sentence has grammar mistakes	<i>The difference of the squares of a number and 6 are 18. Find the number.</i> Here, the noun 'difference' is associated with the auxiliary verb 'are'.
Math constraints	The given numerical values do not lead to a meaningful Mathematical equation	<i>The sum of three consecutive odd integers is 194, what are the integers?</i> This question cannot be solved without changing the values

the generated questions - whether it is more effective for a tutor to correct a generated question, rather than generating a question from scratch. This experiment was conducted only for Sinhala and English, considering mBART-large and mT5-base models. We gave 20 MWPs (10 simple MWPs and 10 Algebraic MWPs) generated by both mBART and mT5 using 50:40 train:test fine-tuning dataset sizes for quarter input seed to 5 university students⁴ who are proficient in English and Sinhala. They were asked to record the time taken to correct each question (refer Table 12 & 13). Then they were given the list of errors we identified in Table 11, and were asked to mark the type of errors they identified. Results of the manual analysis are reported in Tables 14. Note that one generated question may contain more than one type of error.

Table 12: Time taken for a human to correct Simple MWPs (reported in minutes). TTE: Time to Edit 10 generated MWPs, TTG: Time To Generate 10 MWPs

	TTG		TTE		mBART		mT5	
	SE	SS	SE	SS	SE	SS	SE	SS
	TTE		TTE		TTE		TTE	
T1	18	15	2	2.5	0.5	0.38	0.66	0.66
T2	20	25	2.2	3	0.75	0.45	0.48	0.58
T3	15	17.5	1	1.5	0.55	0.38	0.71	0.51
T4	15	28	2.5	1	0.6	0.83	0.6	0.75
T5	21	26.5	3	2	0.63	0.91	0.45	0.6
Av	17.8	22.4	2.14	2	0.60	0.59	0.58	0.62

Table 13: Human evaluation results for Algebraic MWPs in minutes AE: Algebraic English, AS: Algebraic Sinhala, (Number of minutes taken to Edit 10 generated MWPs)

	mBART		mT5	
	AE	AS	AE	AS
Tutor 1	2	0.66	1.16	2
Tutor 2	0.73	0.65	0.58	0.73
Tutor 3	0.42	0.75	0.83	0.78
Tutor 4	0.9	0.88	1.26	1.41
Tutor 5	1.25	1.08	0.91	0.95
Average	1.06	0.80	0.95	1.17

For English MWPs, mT5 model takes the shortest time to correct. For Sinhala MWPs, mBART

⁴Not the same ones who did the translation evaluation

Table 14: Percentages of different types of errors found in simple MWPs

Errors%	mBART				mT5			
	SE	AE	SS	AS	SE	AE	SS	AS
Co-reference	4	4	6	4	8	2	6	2
Unit	4	1	1	1	2	1	1	1
Spelling	0	0	4	2	2	0	0	2
Grammar	16	12	16	10	8	10	14	10
math constraint %	12	38	22	30	14	22	24	32

model takes the shortest time to correct. Note that all these times are less than what Liyanage and Ranathunga (2020) have reported, who in turn have shown that writing questions from scratch takes considerably more time than text generation from their technique.

Co-reference, unit, spelling and grammar are usually less than 20% even in the worst performing model. However, errors related to Math constraint violations are relatively high. This implies that the pre-trained models do not have sufficient information to capture constraints specific to a domain, which of course is not surprising.

6 Conclusion

We evaluated several multilingual and monolingual pre-trained models for the task of MWP generation considering four factors - the amount of language-specific pre-trained data, amount of fine-tuning data, length of the seed and type of the MWP. We also presented a multi-way parallel dataset for MWP evaluation, which includes two languages under-represented in these pre-trained models. Our results are very promising - even with a small amount of parallel data and a short seed, all the models are capable of producing acceptable results for all the considered languages. Human evaluation showed that a Mathematics tutor can take benefit of this automated MWP generation, as it saves time compared to writing an MWP from scratch.

In this research, we did not specifically focus on how to satisfy Maths constraints in an MWP. The effect of this was shown in human evaluation - the questions had a noticeable number of issues related to Math constraints. Thus in the future, we plan

to focus on constraint-based generation of MWP. A starting point would be the work of Wang et al. (2021), who investigated this problem for MWP generation with GPT-2. A major criticism of the pre-trained models is that they support a very small fraction of languages. Thus we want to investigate how the model performance can be improved in the context of languages not included in the model.

7 Ethical Considerations

We have obtained the permission to republish the baseline (Liyanage and Ranathunga, 2020) datasets. In Dolphin18K dataset (Huang et al., 2016) and al1Arith dataset (Roy and Roth, 2016), they have not mentioned any restrictions on using the data. We cited their papers as requested in their repos. We paid the workers according to the rates defined in our university. We verbally explained the purpose of the dataset and the process they have to follow. Worker information was not collected nor included in the dataset, as this is not relevant to the task. In the fine-tuning process, we only focused on elementary-level MWPs. This dataset is publicly released. It does not have any offensive content, nor specific references to individuals or organizations. Thus the fine-tuning process cannot introduce any additional harmful content to the models. We believe that MWP generation in multiple languages has a long-term positive benefit for school children, and the education sector in general. Thus, the positive impact of this research would outweigh any unforeseen negative impacts it could bring.

8 Acknowledgement

Dataset creation of this project was funded by a Senate Research Committee (SRC) grant of University of Moratuwa (UoM), Sri Lanka. The authors would like to thank the National Language Processing Center (NLPC) of UoM for funding the publication of this paper at INLG.

References

Bed Raj Acharya. 2017. Factors affecting difficulties in learning mathematics by mathematics learners. *International Journal of Elementary Education*, 6(2):8–15.

Andinet Assefa Bekele. 2020. Automatic generation of amharic math word problem and equation. *Journal of Computer and Communications*, 8(8):59–77.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198.

Paweł Budzianowski and Ivan Vulić. 2019. Hello, it’s gpt-2—how can i help you? towards the use of pre-trained language models for task-oriented dialogue systems. *arXiv preprint arXiv:1907.05774*.

Pavel Burnyshev, Valentin Malykh, Andrey Bout, Ekaterina Artemova, and Irina Piontkovskaya. 2021. Single example can improve zero-shot data generation. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 201–211.

Tianyang Cao, Shuang Zeng, Songge Zhao, Mairgup Mansur, and Baobao Chang. 2021. Generating math word problems from equations with topic consistency maintaining and commonsense enforcement. In *International Conference on Artificial Neural Networks*, pages 66–79. Springer.

Yiran Chen, Zhenqiao Song, Xianze Wu, Danqing Wang, Jingjing Xu, Jiase Chen, Hao Zhou, and Lei Li. 2021. Mtg: A benchmarking suite for multilingual text generation. *arXiv preprint arXiv:2108.07140*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Thiago Castro Ferreira, Claire Gardent, Nikolai Ilinykh, Chris van der Lee, Simon Mille, Diego Moussallem, and Anastasia Shimorina. 2020. The 2020 bilingual, bi-directional webnlg+ shared task: Overview and evaluation results (webnlg+ 2020). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 55–76.

Apar Garg, Saiteja Adusumilli, Shanmukha Yenneti, Tapas Badal, Deepak Garg, Vivek Pandey, Abhishek Nigam, Yashu Kant Gupta, Gyan Mittal, and Rahul Agarwal. 2020. News article summarization with pretrained transformer. In *International Advanced Computing Conference*, pages 203–211. Springer.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, et al. 2021. The gem benchmark: Natural language generation, its evaluation and metrics. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 96–120.

- Danqing Huang, Shuming Shi, Chin-Yew Lin, Jian Yin, and Wei-Ying Ma. 2016. How well do computers solve math word problems? large-scale dataset construction and evaluation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 887–896.
- Rik Koncel-Kedziorski, Ioannis Konstas, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2016. A theme-rewriting approach for generating algebra word problems. *arXiv preprint arXiv:1610.06210*.
- Aman Kumar, Himani Shrotriya, Prachi Sahu, Raj Dabre, Ratish Puduppully, Anoop Kunchukuttan, Amogh Mishra, Mitesh M Khapra, and Pratyush Kumar. 2022. Indicnlg suite: Multilingual datasets for diverse nlg tasks in indic languages. *arXiv preprint arXiv:2203.05437*.
- Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499.
- En-Shiun Annie Lee, Sarubi Thillainathan, Shravan Nayak, Surangika Ranathunga, David Ifeoluwa Adelan, Ruisi Su, and Arya McCarthy. 2022. Pre-trained multilingual sequence-to-sequence models: A hope for low-resource language translation? In *Findings of the Association for Computational Linguistics 2022*.
- Jieh-Sheng Lee and Jieh Hsiang. 2020. Patent claim generation by fine-tuning openai gpt-2. *World Patent Information*, 62:101983.
- Leo Leppänen, Myriam Munezero, Mark Granroth-Wilding, and Hannu Toivonen. 2017. Data-driven news generation for automated journalism. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 188–197.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Dayiheng Liu, Yu Yan, Yeyun Gong, Weizhen Qi, Hang Zhang, Jian Jiao, Weizhu Chen, Jie Fu, Linjun Shou, Ming Gong, et al. 2021. Glge: A new general language generation evaluation benchmark. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 408–420.
- Tianqiao Liu, Qian Fang, Wenbiao Ding, and Zitao Liu. 2020. Mathematical word problem generation from commonsense knowledge graph and equations. *arXiv preprint arXiv:2010.06196*.
- Vijini Liyanage and Surangika Ranathunga. 2019. A multi-language platform for generating algebraic mathematical word problems. In *2019 14th Conference on Industrial and Information Systems (ICIIS)*, pages 332–337. IEEE.
- Vijini Liyanage and Surangika Ranathunga. 2020. Multi-lingual mathematical word problem generation using long short term memory networks with enhanced input features. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4709–4716.
- Simon Mille, Anja Belz, Bernd Bohnet, Thiago Castro Ferreira, Yvette Graham, and Leo Wanner. 2020. The third multilingual surface realisation shared task (sr’20): Overview and evaluation results. In *Proceedings of the Third Workshop on Multilingual Surface Realisation*, pages 1–20.
- Ahmadreza Mosallanezhad, Kai Shu, and Huan Liu. 2020. Topic-preserving synthetic news generation: An adversarial deep reinforcement learning approach. *arXiv preprint arXiv:2010.16324*.
- Kumaresh Nandhini and Sadhu Ramakrishnan Balasundaram. 2011. Math word question generation for training the students with learning difficulties. In *Proceedings of the International Conference & Workshop on Emerging Trends in Technology*, pages 206–211.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Oleksandr Polozov, Eleanor O’Rourke, Adam M Smith, Luke Zettlemoyer, Sumit Gulwani, and Zoran Popović. 2015. Personalized mathematical word problem generation. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.
- Leonardo FR Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2021. Investigating pretrained language models for graph-to-text generation. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 211–227.

Melissa Roemmele. 2016. Writing stories with help from recurrent neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Subhro Roy and Dan Roth. 2016. Unit dependency graph and its application to arithmetic word problem solving. *arXiv preprint arXiv:1612.00969*.

Leanne J Rylands and Carmel Coady. 2009. Performance of students with weak mathematics in first-year mathematics and science. *International Journal of Mathematical Education in Science and Technology*, 40(6):741–753.

Bowen Tan, Zichao Yang, Maruan Al-Shedivat, Eric P Xing, and Zhiting Hu. 2020. Progressive generation of long text with pretrained language models. *arXiv preprint arXiv:2006.15720*.

Ryota Tanaka, Kyosuke Nishida, and Sen Yoshida. 2021. Visualmrc: Machine reading comprehension on document images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13878–13888.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

Patrick W Thompson. 1985. Experience, problem solving, and learning mathematics: Considerations in developing mathematics curricula. *Teaching and learning mathematical problem solving: Multiple research perspectives*, pages 189–243.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Ke Wang and Zhendong Su. 2016. Dimensionally guided synthesis of mathematical word problems. In *IJCAI*, pages 2661–2668.

Zichao Wang, Andrew Lan, and Richard Baraniuk. 2021. Math word problem generation with mathematical consistency and problem context constraints. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5986–5999.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Qingyu Zhou and Danqing Huang. 2019. Towards generating math word problems from equations and topics. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 494–503.

A Appendix

Figure 1: Sinhala and Tamil Example MWP

Language	Example
Sinhala	සංඛ්‍යා දෙකක එකතුව විසිනුකක් වන අතර විශාල සංඛ්‍යාවකුඩා සංඛ්‍යාවට වඩා පහක් වැඩිය. මෙම සංඛ්‍යා සොයන්න.
Tamil	இரண்டு எண்களின் கூட்டுத்தொகை இருபத்து மூன்று. பெரிய எண் சிறிய எண்ணை விட ஐந்து அதிகம். இந்த எண்களைக் கண்டறியவும்.

Table 15: Zeroshot result ROUGE score for Sinhala and Tamil

Test Dataset	Train Size	Test Size	mBART		mT5	
			R-1	R-2	R-1	R-2
ES	0	986	0.467	0.342	0.026	0.005
EA	0	1175	0.439	0.322	0.022	0.003
SS	0	986	0.411	0.275	0.013	0.001
SA	0	1175	0.378	0.248	0.010	0.001
TS	0	986	0.423	0.286	0.007	0.001
TA	0	1175	0.363	0.247	0.005	0.001
ES	100	986	0.241	0.172	0.057	0.024
EA	100	1175	0.352	0.129	0.117	0.022
SS	100	986	0.539	0.362	0.156	0.048
SA	100	1175	0.212	0.074	0.050	0.010
TS	100	986	0.494	0.221	0.076	0.018
TA	100	1175	0.411	0.189	0.031	0.001

Table 16: ROUGE score results for different domain train and test sizes

Train Dataset	Train Size	Test Dataset	Test Size	mBART		mT5	
				R-1	R-2	R-1	R-2
SA	1679 (40%)	SS	1580 (50%)	0.354	0.246	0.372	0.249
SS	1264 (40%)	SA	2088 (50%)	0.301	0.193	0.271	0.142
TA	1679 (40%)	TS	1580 (50%)	0.384	0.276	0.467	0.324
TS	1264 (40%)	TA	2088 (50%)	0.355	0.253	0.323	0.209

Table 17: Effect of the fine-tuning Dataset Size reported in ROUGE (for quarter seed length)

Dataset size	Train Size	Test Size	English										Tamil				Sinhala			
			GPT2		BART		T5		mBART		mT5		mBART		mT5		mBART		mT5	
			R-1	R-2	R-1	R-2	R-1	R-2	R-1	R-2	R-1	R-2	R-1	R-2	R-1	R-2	R-1	R-2	R-1	R-2
ALG 4210	3370 (80%)	420 (10%)	0.61	0.44	0.61	0.42	0.66	0.50	0.68	0.53	0.65	0.47	0.56	0.40	0.54	0.36	0.49	0.30	0.48	0.28
	1679 (40%)	420 (10%)	0.60	0.42	0.59	0.39	0.64	0.62	0.63	0.46	0.61	0.43	0.54	0.38	0.53	0.36	0.46	0.28	0.44	0.26
	835 (20%)	420 (10%)	0.59	0.51	0.57	0.38	0.62	0.44	0.57	0.38	0.59	0.40	0.51	0.35	0.50	0.34	0.45	0.27	0.42	0.24
SIM 3160	2530 (80%)	316 (10%)	0.64	0.46	0.66	0.51	0.72	0.58	0.72	0.59	0.71	0.57	0.70	0.56	0.66	0.50	0.67	0.52	0.62	0.47
	1264 (40%)	316 (10%)	0.63	0.45	0.61	0.44	0.68	0.68	0.68	0.54	0.66	0.52	0.65	0.49	0.64	0.47	0.63	0.58	0.56	0.53
	632 (20%)	316 (10%)	0.62	0.45	0.59	0.42	0.66	0.52	0.66	0.51	0.62	0.45	0.64	0.48	0.60	0.44	0.59	0.43	0.53	0.36

Comparing informativeness of an NLG chatbot vs graphical app in diet-information domain

Simone Balloccu

University of Aberdeen, UK
simone.balloccu@abdn.ac.uk

Ehud Reiter

University of Aberdeen, UK
e.reiter@abdn.ac.uk

Abstract

Visual representation of data like charts and tables can be challenging to understand for readers. Previous work showed that combining visualisations with text can improve the communication of insights in static contexts, but little is known about interactive ones. In this work we present an NLG chatbot that processes natural language queries and provides insights through a combination of charts and text. We apply it to nutrition, a domain communication quality is critical. Through crowd-sourced evaluation we compare the informativeness of our chatbot against traditional, static diet-apps. We find that the conversational context significantly improved users understanding of dietary data in various tasks, and that users considered the chatbot as more useful and quick to use than traditional apps.

1 Introduction

Visual representations of data is commonly used to communicate insights to the reader. However, understanding the meaning of charts or other visualisations can be challenged by visual deficit, information context, or just the required cognitive effort. Previous research investigated on generating textual explanations of data and comparing them with visualisations (Gatt et al., 2009; Molina et al., 2011; Gkatzia et al., 2017). Approaches like these are particularly useful in healthcare, where lots of data get produced and communication plays a critical role (Zolnerek and DiMatteo, 2009; Brock et al., 2013). Most of these works showed that combining text and visuals improve users' under-

standing of data but they explored static contexts only, where information is presented in a fixed way and there is no active interaction with the reader. Little is known about the effects of text and charts combination in dynamic scenarios, such as conversational ones. Since chatbots are emerging as tools for healthcare (Zhang et al., 2020), it is important to assess if they can provide better communication than static tools (e.g. e-health apps).

In this work we develop and evaluate an NLG-chatbot that generates insights explanation by combining graphics and text. Using our chatbot, users do not need to explore or interpret data themselves, as they can directly ask what they're looking for and get it, along with explanation. We apply it to diet coaching, a domain where communication quality is critical (Van Dorsten and Lindley, 2008; Savolainen, 2010; Michie et al., 2011) and often overlooked by existing tools (Balloccu et al., 2021; Balloccu and Reiter, 2022). To assess the effectiveness of this approach, we run a human evaluation in which we compare our chatbot with traditional diet apps. Participants were assigned to either our chatbot or an app, and used it to take a 10-point quiz concerning the extraction of insights from a simulated food diary. At the end, participants expressed a feedback on the assigned tool. Results show that using our chatbot led to significantly higher scores compared to using traditional apps, both in general and with regards to particular sub-topics. Feedback analysis also reveal that participants perceived our chatbot as more useful for finding diet problems and quicker to use than traditional diet apps.

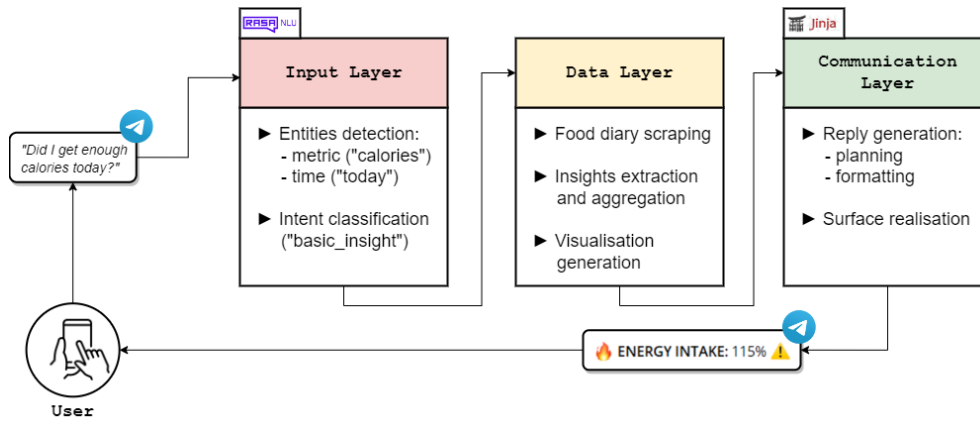


Figure 1: Chatbot architecture and interaction flow.

2 Related work

In this section we recap past research on charts and text combination for insights explanation. We first look at more general work, then move to healthcare and diet-coaching.

2.1 Text vs Graphics in NLG

Previous work investigated how NLG can enhance understanding of data by combining textual content and images. Work on weather data (Gkatzia et al., 2017), showed mixed text and pictures improving decision-making over images alone. Dashboards (Ramos-Soto et al., 2017) benefit from textual explanation of charts as well, as it helps assessing learning in students. Combining charts with explanation of sensors data (Molina et al., 2011) helps insights understanding for general users. Driving reports (Braun et al., 2015) are more helpful if presented as a mix of pictures and text. Healthcare data can also be explained through NLG (Pauws et al., 2019). Experiments in NICU (Law et al., 2005; van der Meulen et al., 2010) suggest that combining charts and text could be the preferred approach by clinicians.

2.2 Text vs Graphics in diet-coaching

Information quality and communication plays a big role in diet (Van Dorsten and Lindley, 2008; Savolainen, 2010; Michie et al., 2011). This applies to apps as well: comprehensibility showed to be a predictor of

prolonged app use (Lee and Cho, 2017). Sub-optimal communication can confuse and demotivate users, leading to early abandonment (Murnane et al., 2015; Mukhtar, 2016). Despite this, diet apps (like MyFitnessPal¹ or FatSecret²) typically come as calorie counters, where users log their meals to obtain insights. These tools adopt very limited textual communication and make extensive use of visualisations that must be interpreted by users themselves (Balloccu and Reiter, 2022). Considering the relationship between numeracy and nutrition literacy (Mulders et al., 2018), this poses a barrier between users and the delivered information. Our previous work (Balloccu et al., 2021) showed similar issues for conversational agents: chatbots adopt fixed educational material (Casas et al., 2018; Stephens et al., 2019; Davis et al., 2020), such as PDFs containing guidelines, and expose lack of reasoning over user queries (Maher et al., 2020). Similarly to apps, chatbots show plain reports, with little to no feedback on goals, progress or mistakes (Casas et al., 2018; Prasetyo et al., 2020).

3 NLG chatbot to improve communication quality

Our chatbot consists of an Input Layer for users' input understanding; a Data Layer that extracts insights and generates visualisations; a Communication Layer that per-

¹www.myfitnesspal.com

²<https://www.fatsecret.com/>

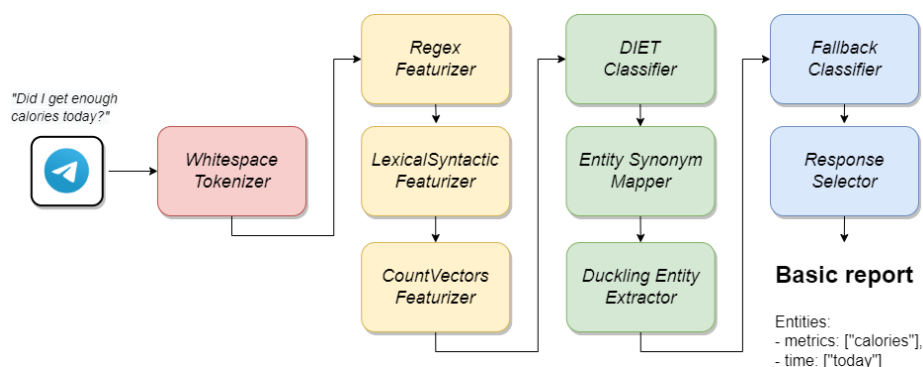


Figure 2: Overview of the NLU pipeline.

forms planning and surface realisation (Figure 1). We use RASA Open Source 2.0³ as the main infrastructure for the entire system, and exploit its NLU component (Figure 2) for the Input layer; the Data Layer adopts a custom data analysis logic; the Communication Layer adopts rule-based NLG and variable templates (through Jinja 3.0⁴).

We adopt an hybrid architecture: we use machine-learning for NLU but restrict text generation to rules. This is mainly for two reasons: 1) diet domain imposes strict accuracy requirements that cannot be met by current E2E NLG (Thomson and Reiter, 2020; van Miltenburg et al., 2021) and 2) to the best of our knowledge, there is no publicly available diet-coaching corpus which can be used to train or fine-tune generative models. On the other hand, machine-learning offers good generalisation for NLU with the only risk being unexpected inputs or failure in intent classification.

We model two main interactions into the chatbot: basic reports and comparisons (Figure 3). Basic reports show insights about a single time frame, either as brief information on energy and nutrients balance or combinations of charts and text. Comparisons extend basic reports to multiple time frames by informing users about progress (e.g. improved intake; changes in food choices etc..). For each request, users can specify metrics (calories and five nutrients: carbohydrates, protein, fat, sugar and sodium) and time (de-

tected via Duckling Entity extractor⁵). This approach offers more flexibility than traditional apps, that typically aggregates all the metrics in a single section (e.g. a table) and present pre-defined comparisons (e.g. every month).

3.1 Explanation through text and charts

Users can access two typologies of insights: basic and advanced. Basic insights show energy and nutrients intake (see Figure 3) as brief textual messages. This is thought for users that need a quick glance at their data. Advanced insights deliver more information and are presented as a combination of text and charts. Users can obtain the following advanced insights (Figure 4):

1. **Intake analysis:** reasons and explains intakes with regards to user goals.
2. **Trend and consistency:** detects if trends match recommended changes in diet (e.g. getting less calories to fix an excess) and checks intake consistency (maintaining a stable intake across days).
3. **Food analysis:** reasons and explains intakes at food level, by showing which food has the biggest impact.

Advanced insights naturally extend to comparisons as well (Figure 4). To let both novice users (that need supervision) and advanced ones access advanced insights, they can be obtained in two ways (Figure 5):

³<https://rasa.com/docs/rasa/>

⁴<https://jinja.palletsprojects.com/en/3.0.x/>

⁵<https://duckling.wit.ai/>

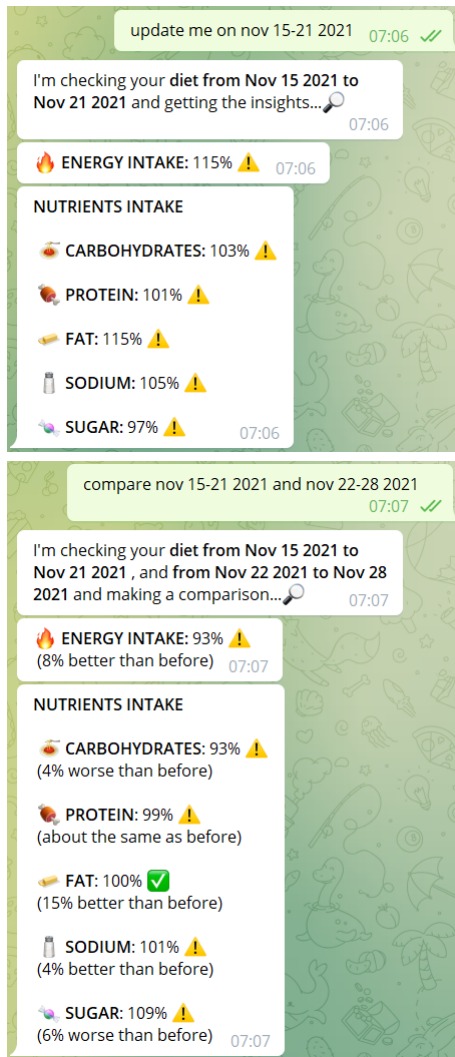


Figure 3: Basic report and comparison as presented by the chatbot.

1. **Guided navigation:** through generic queries (e.g. "tell me more about this" or "anything else?"). Following this trigger, the chatbot presents a button interface for each available advanced insight. Buttons can be checked and unchecked to obtain only those insights that are of interest.
2. **Natural language query:** by directly asking for specific insights and metrics. This can be done by specifying a particular insight (e.g. "food" or "intake") on a specific period.

For both interactions, users can specify one or more metrics.

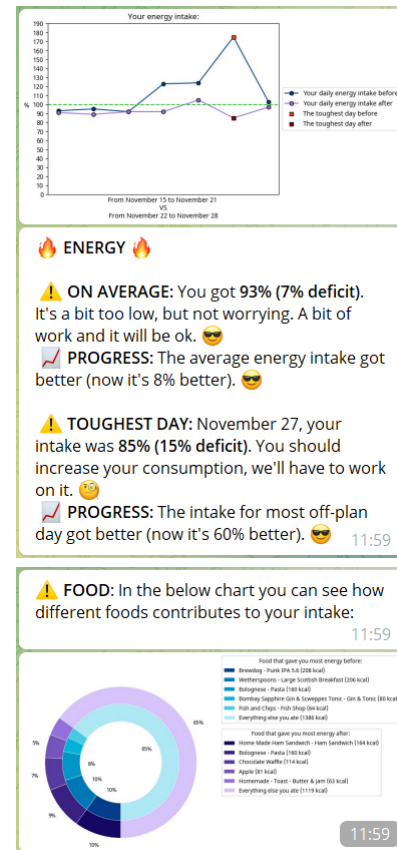


Figure 4: Example of advanced insights (intake and food analysis) for comparisons.

3.2 Other features

We implement a number of supplementary best practices (Ferman, 2018) to further improve usability and clarity. The chatbot actively provides feedback for each input (while informing users on the pending task); adopts emojis to make insights more understandable; splits the content in multiple messages and introduce a dynamic delay between them to avoid flooding.

4 Experiment setup

We deploy our chatbot on Telegram Bot API⁶ and compare its informativeness with traditional diet apps. We gather our test population (**workers**) through crowd-sourcing on Amazon Mechanical Turk⁷. Details of recruitment, pay and sanity checks are available in the Appendix A. We choose to compare our

⁶<https://core.telegram.org/bots/api>

⁷<https://www.mturk.com/worker/help>

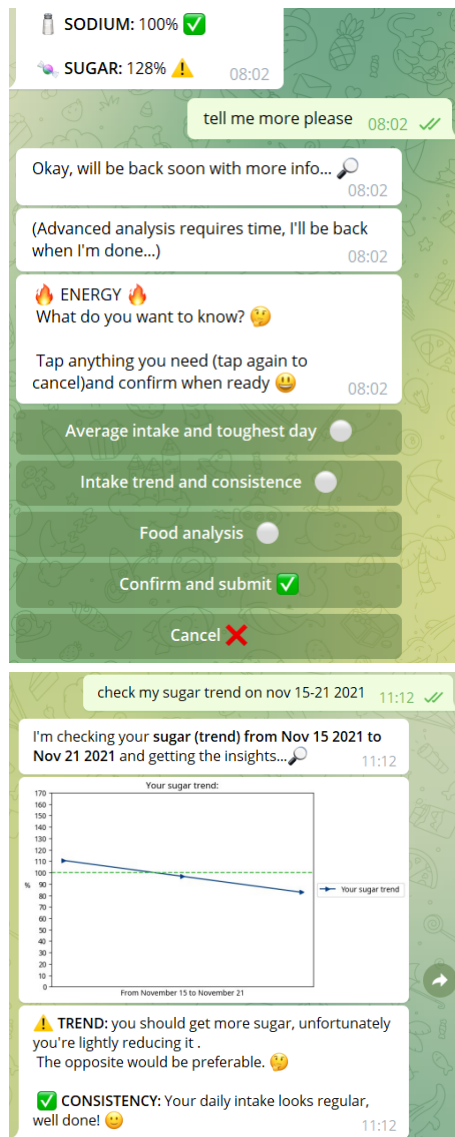


Figure 5: Obtaining advanced insights: Guided navigation with buttons (top) VS Natural language uery about trend and consistency (bottom).

chatbot with MyfitnessPal⁸ (MFP) and Fat-Secret⁹ (FS). An example of the two apps UI can be seen in Figure 6. We choose these two apps based on their popularity and downloads number on the Apple and Android app stores. We do not compare against any dieting chatbot as none of those present in literature is publicly available.

4.1 Measuring informativeness

Aiming at communication improvement, we need to find a measure to capture whether

⁸<https://www.myfitnesspal.com/>

⁹<https://www.fatsecret.com/>

one specific tool performs better than others. From communication theory (Webster and Morris, 2019) we adopt the concept of "informativeness", defined as "how successfully a person is able to convey an intended message". We extend this definition to diet systems as "how successfully a tool is able to convey an intended message". To capture informativeness we create a ten questions quiz regarding diet analysis (a sample is provided in Appendix B). The quiz consists of 4 macro-tasks:

1. **Day analysis:** understanding if calories and carbohydrates are balanced on a single day (2pts).
2. **Food analysis:** understanding what food provided most calories and fat on a single day, along with quantities (4pts).
3. **Week analysis:** understanding if calories and carbohydrates are balanced across a week (2pts).
4. **Weeks comparison:** understanding if, by comparing two weeks, calories and carbohydrates improved or worsened (2pts).

Each question is worth 1 point, for a total of 10 points. We choose to develop a custom quiz because no available questionnaire can be used evaluate the informativeness of a diet-coaching tool. In creating it, we analyse existing apps and all the information that they deliver; we incorporate experts recommendations from previous surveys (Vasiloglou et al., 2020); we consider the theoretical constructs of self-regulation (Zahry et al., 2016), with a particular focus on the measure of informativeness. We avoid evaluating "trend and consistency" feature for fairness, as apps don't offer a way for the user to infer such information without long and tedious calculations.

Workers were randomly assigned to either our chatbot, MFP or FS, each of which was pre-filled with a simulated food diary (none of the data belonged to the users) consisting of 2 weeks of logged meals. We obtained

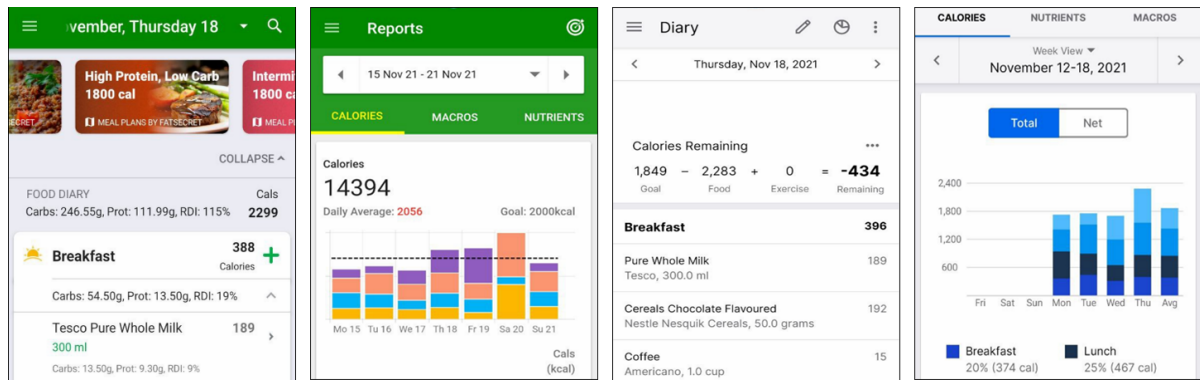


Figure 6: Food Diary and nutrition reports as showed to the user in FatSecret (left) and MyFitnessPal UI (right).

$n=27$ workers assigned to our chatbot; $n=31$ workers to MFP; $n=29$ workers to FS. Besides the tool itself, workers were provided with a PDF guide on how to use it and a glossary explaining the meaning of the terms used in the quiz. Each worker took the quiz and was asked to answer the questions to the best of their knowledge by using the tool. Through the quiz we test the following hypothesis:

Hypothesis 1 (H1): *Chatbot workers scored higher on informativeness quiz than MFP or FS workers.*

4.2 Measuring nutrition literacy

Previous research highlighted the importance of nutrition literacy in dieting (Michie et al., 2011), so we analyse its impact on our experiment. We also analyse if our chatbot communication can reduce the score gap between different literacy levels. Before taking the quiz, each worker completed Pfizer's Newest-Vital-Sign (NVS) (Weiss et al., 2005; Powers et al., 2010), consisting of 6 questions (each one worth 1 point) regarding an ice-cream label. NVS scores are grouped in ranges: 0-1 refers to "high likelihood of limited literacy", 2-3 refers to "possibility of limited literacy"; 4-6 refers to "adequate literacy". We compare NVS scores with quiz scores to test the following hypothesis:

Hypothesis 2 (H2): *There was a positive correlation between NVS score and quiz score in our experiment, but not for chatbot workers.*

4.3 Measuring perception of the tool and past experience

Finally, we inspect workers opinion on the tool they used. We ask each worker to rate the tool under different characteristics (see Figure 8) through Likert-5 scale. Through this approach we test the following hypothesis:

Hypothesis 3 (H3): *Our chatbot received higher ratings across the proposed questions.*

Finally, we ask workers to specify whether they had past experience with dieting tools (including the one they were assigned to) and to specify how often they used it (often; occasionally; rarely; never).

5 Results analysis

For variance analysis, we adopt One-Way ANOVA and Tukey's post-hoc test (replaced respectively by Kruskal-Wallis test and Dunn's post-hoc test if ANOVA's normality requirement is not met). To test variable dependence we adopt Chi-squared test and Bonferroni's post-hoc test. For correlation test we adopt Pearson correlation (substituted by Spearman correlation if Pearson's normality requirement is not met).

5.1 Preliminary checks

Before analysing results, we verify nutrition literacy uniformity across our population, to ensure that none of the groups contained mostly workers with high/low nutrition literacy. We discover that nutrition literacy

Topic	Average score			Score differences		
	CB	FS	MFP	CB-FS	CB-MFP	MFP-FS
Overall (10pt)	6.65	4.13	5.22	+2.52**	+1.43	+1.09
Day analysis (2pt)	1.15	0.76	1.32	+0.40	-0.16	+0.56
Food analysis (4pt)	2.85	2.14	0.91	+0.71	+1.94***	-1.23*
Week analysis (2pt)	1.35	0.66	1.05	+0.70**	+0.30	+0.39
Weeks comparison (2pt)	1.31	0.59	1.14	+0.72**	+0.17	+0.55**

Table 1: Results from informativeness quiz. On the left side: average scores, overall and for specific tasks. Highest score for each category are in bold. On the right side: score differences between tools. Green is for higher scores, red is for lower score. CB = Chatbot; MFP = MyFitnessPal; FS = FatSecret. Significance: * for $p < 0.05$; ** for $p < 0.01$; *** for $p < 0.001$.

NVS class	Workers per class		
	CB	FS	MFP
LOW (0-1pt)	1	0	9
MID (2-3pt)	5	3	5
HIGH (4-6pt)	21	26	17

Table 2: Distribution of nutrition literacy for our population. CB = Chatbot; MFP = MyFitnessPal; FS = FatSecret.

distribution is unbalanced among apps, with the majority of workers with low nutrition literacy assigned to MFP sample, none to FS only one to our chatbot (see Table 2). We re-balance the samples by removing all the such workers workers. This limits our inspections on nutrition literacy but keeps the comparison fair. From now on, all results will refer to the re-balanced sample unless otherwise specified. We also check for meaningful difference in workers past experience with diet tools, but find none neither in general ($p = 0.47$) and by considering only those workers who had past experience and ($p = 0.27$).

5.2 Quiz scores

We first check total and per-task quiz scores (see Table 1). We find that, overall, the highest average score was reached by chatbot workers. The difference was statistically significant when compared to FS workers. Regardless of the group, average scores were low, not going much higher than 6/10. We consider this as a further confirmation of how hard understanding dietary insights is for the average user, especially in our context where data was simulated. By inspecting individual quiz tasks, we see that chatbot workers scored significantly higher in week analysis and comparison than FS workers,

and in food analysis than MFP workers. We also find that MFP workers scored significantly higher than FS workers when comparing weeks, while the opposite happened for food analysis. Overall, chatbot workers always scored the highest score in every case, except for the day analysis, where MFP workers scores were slightly higher.

Next we look at the percentage of correct answers to check if any of the tools were associated with reaching specific scores (e.g. maximum points or 0 points). First, we find that our chatbot was positively associated ($p = 0.0001$) with an overall score of 9/10 points. This tells us that the chatbot made it easier to reach higher scores in general. We then proceed to analyse individual quiz tasks (Figure 7). Our chatbot was positively associated with maximum score in food analysis and week analysis. For chatbot workers it was easier understanding food details and insights based on aggregation in general. It was also negatively associated ($p = 0.001$) with 0 points in weeks comparison. In fact, every chatbot worker managed to answer at least one of the two questions about comparison right. Interestingly, we find the opposite for FS, that was positively associated with scoring 0 points in weeks comparison. This tells us that FS workers struggled considerably in this task. Lastly, using MFP was negatively associated with maximum score in food analysis: understanding food details was one of the hardest tasks with MFP.

5.3 Nutrition Literacy effect on scores

We check if nutrition literacy influenced quiz score. In here we discover a discrepancy between the balanced and unbalanced sample. MFP workers show a significant difference

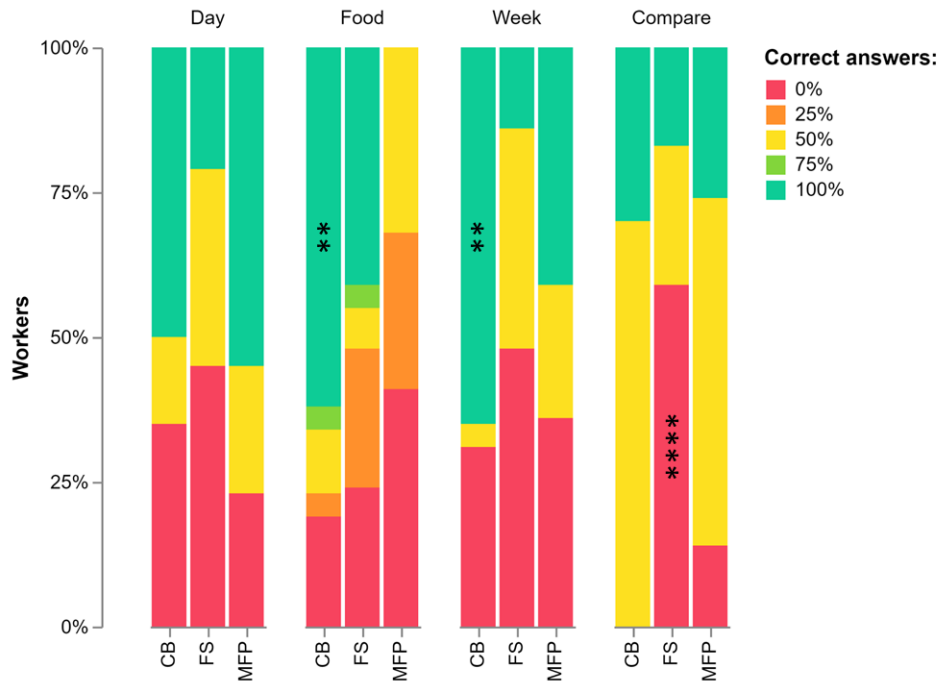


Figure 7: Percentage of correct answers by task, for each tool. CB = Chatbot; FS = FatSecret; MFP = MyFitnessPal. For day, week analysis and comparisons (2-points) we check no right answer (0%), 1 right answer out of 2 (50%) and all right answers (100%). For food analysis (4 points), we check quarters as well. Significance: * for $p < 0.05$; ** for $p < 0.01$; *** for $p < 0.001$; **** for $p < 0.0001$.

($p = 0.03$) in scores between high and low nutrition literacy. By re-balancing the sample, we lose this significance. We also discover a moderate correlation ($\rho = 0.48, p = 0.02$) between nutrition literacy and quiz score for MFP workers, even after balancing the samples.

5.4 Users perception of the tool

Finally, we check workers feedback (see Figure 8). We notice a generally positive evaluation for every tool, with the chatbot getting an higher amount of "Agree" ratings across every question. By single-item analysis, our chatbot was positively associated with "Agree" in Q1 ($p = 0.01$), where it also shows a better mode value than the other tools. Chatbot workers felt it more useful for finding problems in the food diary. We also find a better mode than both apps in Q3, meaning that workers found it to be quicker to use. This result in particular is unexpected considering that there was no significant difference in the quiz execution time ($p = 0.22$). It could be that using natural language in our chatbot was felt as faster

than navigating through different app sections. No app showed better mode than our chatbot in any question. Finally, it is interesting to notice that FS scored higher than MFP in Q5 despite being the tool with the lowest scores across every task except food analysis.

6 Discussion

From quiz results, chatbot workers scored the highest in informativeness across every scenario except for a slight advantage of MFP in day analysis. In multiple contexts, the difference with MFP and FS was statistically significant. We also found that using the chatbot was associated with higher completion rate in different tasks, and very high overall scores like 9/10. With these results we confirm H1. We could not inspect nutrition literacy properly, as the different samples were too unbalanced and introducing low-literate workers would have made the comparison between MFP and our chatbot unfair. We saw a relationship between lower nutrition literacy and quiz scores, but isolated to MFP workers, and could not verify

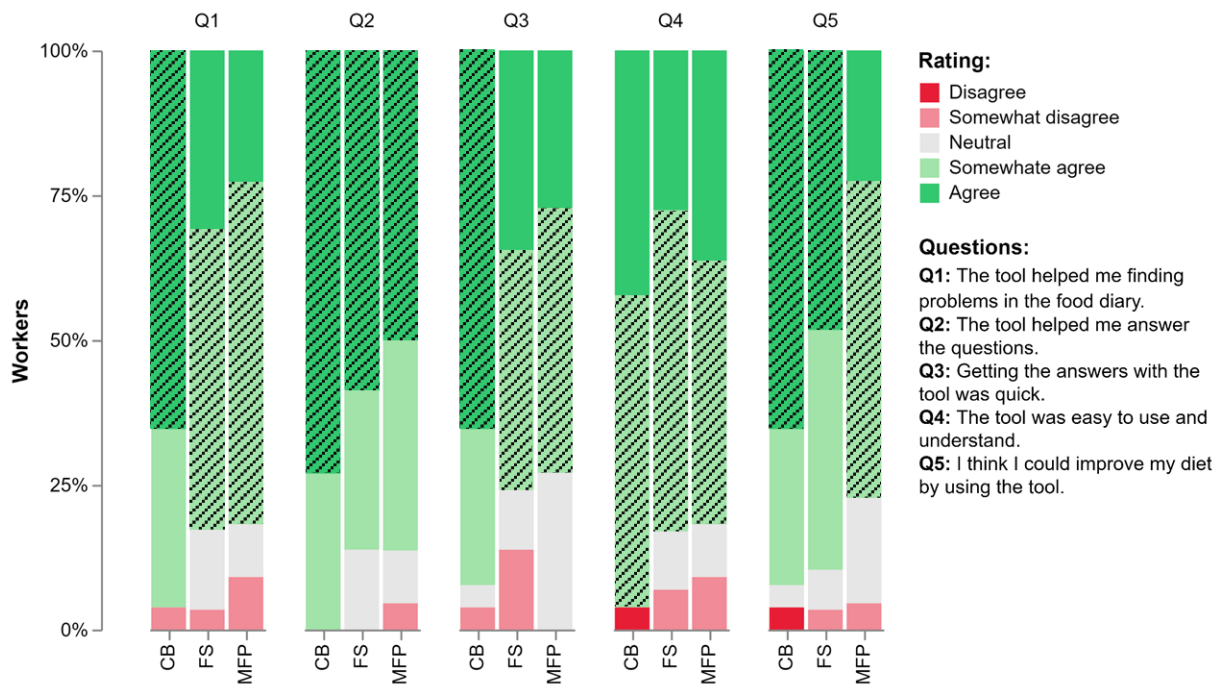


Figure 8: Feedback from users, based on used tool. CB = Chatbot; FS = FatSecret; MFP = MyFitnessPal. Lined bars indicate the mode for each question.

it across the whole population. With these results we neither confirm or reject H2 because of the lack of data. Looking at feedback, we found out that our chatbot received a higher amount of "Agree" ratings across every question. It was also the only tool that showed association with maximum usefulness in finding diet problems. By analysing the mode of each question, we discovered that our chatbot was evaluated as quicker to use than the other apps. We also see that, unlike MFP and FS, it never showed a lower mode than any other tool. With these results we confirm H3.

7 Conclusion and future developments

In this work we evaluated the combination of charts and textual explanation for diet coaching, in the conversational scenario. We implemented an NLG-chatbot that understands natural language input and returns dietary insights as a combination of textual explanations and visualisations. We compared the chatbot with traditional static diet apps by inspecting informativeness and user feedback. Results shows that the combination of visuals and text efficiently delivers infor-

mation in diet-coaching, and makes it more understandable. Improved informativeness could play a critical role in diet outcome. Feedback was generally more positive for the chatbot, meaning that it can be a valid tool for diet-coaching, potentially substituting static apps.

For future work we plan to investigate if our approach can lead to actual learning from the user, for example through spaced repetition (Ausubel and Youssef, 1965; Tabibian et al., 2019) that can positively affect users' forgetting curve (Ebbinghaus, 2013). We also commit on addressing the limits of our setup, to properly inspect the relationship between nutrition literacy and informativeness. We also plan to inspect more personalised approaches to information tailoring, namely by considering users' stress and emotional state that showed to be promising research directions (Balloccu et al., 2020; Balloccu and Reiter, 2022). Lastly, we consider this result as a sign of the maturity of our approach and we plan to run a trial to measure its effect on diet-coaching (e.g. weight control).

References

- David P Ausubel and Mohamed Youssef. 1965. [The effect of spaced repetition on meaningful retention](#). *The Journal of General Psychology*, 73(1):147–150.
- Simone Balloccu and Ehud Reiter. 2022. [Beyond calories: evaluating how tailored communication reduces emotional load in diet-coaching](#). In *Proceedings of the 2nd Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 42–53, Dublin, Ireland. Association for Computational Linguistics.
- Simone Balloccu, Ehud Reiter, Matteo G. Collu, Federico Sanna, Manuela Sanguinetti, and Maurizio Atzori. 2021. [Unaddressed Challenges in Persuasive Dieting Chatbots](#), page 392–395. Association for Computing Machinery, New York, NY, USA.
- Simone Balloccu, Ehud Reiter, Alexandra Johnstone, and Claire Fyfe. 2020. [How are you? introducing stress-based text tailoring](#). In *Proceedings of the Workshop on Intelligent Information Processing and Natural Language Generation*, pages 62–70, Santiago de Compostela, Spain. Association for Computational Linguistics.
- Daniel Braun, Ehud Reiter, and Advait Sidharthan. 2015. [Creating textual driver feedback from telemetric data](#). In *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*, pages 156–165, Brighton, UK. Association for Computational Linguistics.
- Douglas Brock, Erin Abu-Rish, Chia-Ru Chiu, Dana Hammer, Sharon Wilson, Linda Vorvick, Katherine Blondon, Douglas Schaad, Debra Liner, and Brenda Zierler. 2013. [Republished: Interprofessional education in team communication: working together to improve patient safety](#). *Postgraduate medical journal*, 89(1057):642–651.
- Jacky Casas, Elena Mugellini, and Omar Abou Khaled. 2018. [Food diary coaching chatbot](#). In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers, UbiComp '18*, page 1676–1680, New York, NY, USA. Association for Computing Machinery.
- Courtney R Davis, Karen J Murphy, Rachel G Curtis, and Carol A Maher. 2020. [A process evaluation examining the performance, adherence, and acceptability of a physical activity and diet artificial intelligence virtual health assistant](#). *International journal of environmental research and public health*, 17(23):9137.
- Hermann Ebbinghaus. 2013. [Memory: A contribution to experimental psychology](#). *Annals of neurosciences*, 20(4):155.
- Maria Ferman. 2018. [Towards best practices for chatbots](#).
- Albert Gatt, François Portet, Ehud Reiter, Jim Hunter, Saad Mahamood, Wendy Moncur, and Somayajulu Sripada. 2009. [From data to text in the neonatal intensive care unit: Using nlg technology for decision support and information management](#). *AI Commun.*, 22(3):153–186.
- Dimitra Gkatzia, Oliver Lemon, and Verena Rieser. 2017. [Data-to-text generation improves decision-making under uncertainty](#). *IEEE Computational Intelligence Magazine*, 12(3):10–17.
- Anna S Law, Yvonne Freer, Jim Hunter, Robert H Logie, Neil McIntosh, and John Quinn. 2005. [A comparison of graphical and textual presentations of time series data to support medical decision making in the neonatal intensive care unit](#). *Journal of clinical monitoring and computing*, 19(3):183–194.
- H Erin Lee and Jaehee Cho. 2017. [What motivates users to continue using diet and fitness apps? application of the uses and gratifications approach](#). *Health communication*, 32(12):1445–1453.
- Carol Ann Maher, Courtney Rose Davis, Rachel Grace Curtis, Camille Elizabeth Short, and Karen Joy Murphy. 2020. [A physical activity and diet program delivered by artificially intelligent virtual health coach: Proof-of-concept study](#). *JMIR mHealth and uHealth*, 8(7):e17558.
- Susan Michie, Maartje M Van Stralen, and Robert West. 2011. [The behaviour change wheel: a new method for characterising and designing behaviour change interventions](#). *Implementation science*, 6(1):1–12.
- Martin Molina, Amanda Stent, and Enrique Parodi. 2011. [Generating automated news to explain the meaning of sensor data](#). In *Advances in Intelligent Data Analysis X*, pages 282–293, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Hamid Mukhtar. 2016. [Using persuasive recommendations in wellness applications based upon user activities](#). *International Journal of Advanced Computer Science and Applications*, 7(8).
- Maria D.G.H. Mulders, O. Corneille, and O. Klein. 2018. [Label reading, numeracy and food nutrition involvement](#). *Appetite*, 128:214–222.

- Elizabeth L. Murnane, David Huffaker, and Gueorgi Kossinets. 2015. [Mobile health apps: Adoption, adherence, and abandonment](#). In *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*, UbiComp/ISWC'15 Adjunct, page 261–264, New York, NY, USA. Association for Computing Machinery.
- Steffen Pauws, Albert Gatt, Emiel Krahmer, and Ehud Reiter. 2019. [Making Effective Use of Healthcare Data Using Data-to-Text Technology](#), pages 119–145. Springer International Publishing, Cham.
- Benjamin J Powers, Jane V Trinh, and Hayden B Bosworth. 2010. [Can this patient read and understand written health information?](#) *Jama*, 304(1):76–84.
- Philips Kokoh Prasetyo, Palakorn Achananuparp, and Ee-Peng Lim. 2020. [Foodbot: A Goal-Oriented Just-in-Time Healthy Eating Interventions Chatbot](#), page 436–439. Association for Computing Machinery, New York, NY, USA.
- Alejandro Ramos-Soto, Borja Vazquez-Barreiros, Alberto Bugarín, Adriana Gewerc, and Senen Barro. 2017. [Evaluation of a data-to-text system for verbalizing a learning analytics dashboard](#). *International Journal of Intelligent Systems*, 32(2):177–193.
- Reijo Savolainen. 2010. [Dietary blogs as sites of informational and emotional support](#).
- Taylor N Stephens, Angela Joerin, Michiel Rauws, and Lloyd N Werk. 2019. [Feasibility of pediatric obesity and prediabetes treatment support through tess, the ai behavioral coaching chatbot](#). *Translational behavioral medicine*, 9(3):440–447.
- Behzad Tabibian, Utkarsh Upadhyay, Abir De, Ali Zarezade, Bernhard Schölkopf, and Manuel Gomez-Rodriguez. 2019. [Enhancing human learning via spaced repetition optimization](#). *Proceedings of the National Academy of Sciences*, 116(10):3988–3993.
- Craig Thomson and Ehud Reiter. 2020. [A gold standard methodology for evaluating accuracy in data-to-text systems](#). In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 158–168, Dublin, Ireland. Association for Computational Linguistics.
- Marian van der Meulen, Robert H. Logie, Yvonne Freer, Cindy Sykes, Neil McIntosh, and Jim Hunter. 2010. [When a graph is poorer than 100 words: A comparison of computerised natural language generation, human generated descriptions and graphical displays in neonatal intensive care](#). *Applied Cognitive Psychology*, 24(1):77–89.
- Brent Van Dorsten and Emily M Lindley. 2008. [Cognitive and behavioral approaches in the treatment of obesity](#). *Endocrinology and metabolism clinics of North America*, 37(4):905–922.
- Emiel van Miltenburg, Miruna Clinciu, Ondřej Dušek, Dimitra Gkatzia, Stephanie Inglis, Leo Leppänen, Saad Mahamood, Emma Manning, Stephanie Schoch, Craig Thomson, and Luou Wen. 2021. [Underreporting of errors in NLG output, and what to do about it](#). In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 140–153, Aberdeen, Scotland, UK. Association for Computational Linguistics.
- Maria F. Vasiloglou, Stergios Christodoulidis, Emilie Reber, Thomai Stathopoulou, Ya Lu, Zeno Stanga, and Stavroula Mougiakakou. 2020. [What healthcare professionals think of “nutrition amp; diet” apps: An international survey](#). *Nutrients*, 12(8).
- Janet Webster and Julie Morris. 2019. [Communicative informativeness in aphasia: Investigating the relationship between linguistic and perceptual measures](#). *American Journal of Speech-Language Pathology*, 28(3):1115–1126.
- Barry D Weiss, Mary Z Mays, William Martz, Kelley Merriam Castro, Darren A DeWalt, Michael P Pignone, Joy Mockbee, and Frank A Hale. 2005. [Quick assessment of literacy in primary care: the newest vital sign](#). *The Annals of Family Medicine*, 3(6):514–522.
- Nagwan R Zahry, Ying Cheng, and Wei Peng. 2016. [Content analysis of diet-related mobile apps: A self-regulation perspective](#). *Health Communication*, 31(10):1301–1310.
- Jingwen Zhang, Yoo Jung Oh, Patrick Lange, Zhou Yu, and Yoshimi Fukuoka. 2020. [Artificial intelligence chatbot behavior change model for designing artificial intelligence chatbots to promote physical activity and a healthy diet: Viewpoint](#). *J Med Internet Res*, 22(9):e22845.
- Kelly B Haskard Zolnieriek and M Robin DiMatteo. 2009. [Physician communication and patient adherence to treatment: a meta-analysis](#). *Medical care*, 47(8):826.

A Ethics

This section sums up the procedure we adopted to ensure the ethical compliance of our experiment.

A.1 Preliminary review

Before starting the experiment, procedure and materials were carefully reviewed by our institution Ethics Board (omitted for the sake of double-blinded review). Our experiment proposal was accepted without major revisions.

A.2 Platforms

For the quiz, we adopted Microsoft Forms¹⁰ because of its compliance with GDPR policy. For hiring, we used Amazon Mechanical Turk. No recruitment qualification was specified, beside custom ones to prevent the same worker from submitting multiple HITs. Participants were showed a consent form containing all the information regarding the experiment procedure. They were also informed about the requirements that had to be satisfied to obtain the remuneration. All worker had to confirm their acceptance of these conditions (through checkboxes) in order to proceed with the experiment. Workers were given an email contact in case of problems during the experiment.

A.3 Pay and workload

Before launching the experiment, we verified the average completion time with 10 test users. The average result for completing the whole experiment (reading information; downloading and setting up material; taking NVS; taking the quiz; expressing the feedback) was 20 minutes. We gave each worker 45 minutes, and paid 15USD for the HIT. Workers were informed that if they ran out of time on Mturk they could just finish the quiz (on Microsoft Forms web platform) and contact us through the provided email address to still get paid.

¹⁰<https://forms.office.com/>

A.4 HITs sanity checks

We received a total amount of 250 applications for our task. Most of these application were fraudulent, with random answers or unrealistic completion times. In order to recognise legit HITs we set up multiple sanity-checks, both in general and depending on the tool each worker was assigned.

A.4.1 Global sanity checks

To check on the attention of workers during Pfizer's NVS, a fake price was added to the ice-cream label. Consequently, we added a (non scored) question to the form, asking "what's the price of the ice-cream?". Moreover, each worker received a completion code that they had to submit on Mechanical Turk platform after completing all the tasks.

A.4.2 Sanity check for chatbot worker

The chatbot was programmed to accept some custom queries that led to specific answers. The workers were asked, at multiple times, to trigger one of these query. We manually checked the answers for HITs, in order to verify whether workers actually used the chatbot. In addition, conversations were logged and anonymised, and the provided WorkerID was used to track down specific workers and verify the sanity of interaction.

A.4.3 Sanity check for FS and MFP worker

To verify that workers actually used the diet apps they were asked to provide a description of the app logo, and to check which particular food (among three alternatives) could be seen in a specified day. As this tasks are subjective and could be failed by legit workers who struggled to use the app, each HIT was manually evaluated to avoid unfair treatment.

B Appendix A: Quiz sample

Evaluating the informativeness of various diet-coaching tools

* Required

Introduction

Please read the following instructions carefully before proceeding.

What is this experiment about?

This research aims at evaluating whether common diet-coaching apps are **informative** for users.

In other words, how easy it is for users to find the information they need and, most importantly, how comprehensible they are.

What will I have to do?

For this experiment, we ask you to do 3 main tasks.

1. **Preliminary form:** during this step, you'll be asked to answer a short form (**5-6 questions**) regarding nutrition. This will involve extracting information from a sample nutritional label and reasoning about them.
2. **Main form:** following the completion of the previous point, you'll be assigned to a tool (a diet-coaching app). You'll receive instructions on **how to download (through Play/App Store), install and use the app** on your phone. Each app has been pre-compiled with food diaries (imagine this as someone else record of what they ate). You will be asked to explore this data to answer **10 questions**.
3. **Final feedback:** finally, we will ask you to give us your opinion on the overall experiment (**7 questions**), with a particular focus on the tool you used in step 2. You'll be asked for your **worker ID** and be given a **completion code**. Return it to us to process your HIT.

Total time for doing this experiment should be between **30-45 minutes**.

Additional details (1/2)

Some important things to keep in mind:

1. You'll need to **install and use** the assigned app on your phone (Android/IOS) to complete the experiment.
Failure in complying with this requirement will cause **HIT invalidation**.
2. The experiment is monitored. Fraudulent behaviour such as **completing the form without reading the questions** or **giving random answers** will be detected and will result in the invalidation of your HIT.
3. Note that the previous points does not apply to the cases in which, **despite using the app, you're still not able to give an answer**. Regardless of the amount of correct answer you give, **you will still receive your remuneration**.
4. You will be assigned to **only one app** for this experiment. You won't have to repeat it multiple times.
5. Most of the apps don't require any registration: we'll give you login credentials (username and password).
In only one case, the app will require your phone number for access. We won't be able to see or access this as it is a chat app (**Telegram**).
6. You don't have to keep the app installed after the experiment. You can uninstall it immediately when done.
7. None of the apps have been developed by us and therefore **we won't receive any data except the form answers**.
8. As said before, your assigned app will show you some data regarding food and meals across different days.
Please only read the data, **avoid changing, deleting or altering that data in any way**.
Data alteration/augmentation will result in experiment invalidation (**and HIT invalidation for MTurkers**)
9. In any case, **no data (outside of form answers) will be gathered**.
10. Should you change your mind, you can withdraw from the experiment at any given stage and without giving a reason, until the point in which data analysis shall be done with your (completed) results

Additional details (2/2)

Data management and storage

No personal data about you shall be collected or stored beside the data which will be put in the forms. All your answers will be anonymously and safely stored in devices belonging to University of Aberdeen. None of these data shall be released to the public.

Confidentiality and anonymity

Raw data and the identity of participants will not be released to anyone outside the research team. The data you provide will be analysed and may be used in publications, dissertations, reports or presentations derived from the research project, but this will be done in such a way that your identity is not disclosed.

Consent

If you agree to take part in the research, you will be asked to indicate your consent by ticking the following checkboxes.

Risk

We foresee no risk for any participant involved.

Sponsor

This research is being funded by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 812882.

1

Question *

- I confirm that the research project "**Evaluating the informativeness of various diet-coaching tools**" has been explained to me. I have had the opportunity to ask questions about the project and have had these answered satisfactorily.

2

Question *

- I consent to the material I contribute being used to generate insights for the research project "**Evaluating the informativeness of various diet-coaching tools**".

3

Question *

- I understand that my participation in this research is voluntary and that I may withdraw from the project at any time (until the point of data analysis) without providing a reason. **I understand that (for MTurkers) withdrawal will invalidate my HIT.**

4

Question *

- I consent to allow the **fully anonymised** data to be used for future publications and other **scholarly** means of disseminating the findings from the research project.

5

Question *

- I understand that the information/data acquired will be **securely** stored by researchers, but that **appropriately** anonymised data may in future be made **available** to others for research purposes. I understand that the University may publish **appropriately** anonymised data in its research repository for verification purposes and to make it **accessible** to researchers and other research users.

Preliminary form (1/3)

In this first form, we ask you to answer some questions related to **the nutritional label displayed below**. Answer to the best of your knowledge.

Please do not seek help from anyone else to complete this form. The aim is not to score maximum points at any cost. None of your answers will be shared with anyone and your identity will be kept anonymous.

Additional help:

- You are allowed to use a calculator if you would like to.
- You do not need any app for this part of the experiment.

6

*

Nutrition Facts		
Serving Size		½ cup
Servings per container		4
Amount per serving		
Calories	250	Fat Cal 120
		%DV
Total Fat	13g	20%
Sat Fat	9g	40%
Cholesterol	28mg	12%
Sodium	55mg	2%
Total Carbohydrate	30g	12%
Dietary Fiber	2g	
Sugars	23g	
Protein	4g	8%
<small>*Percentage Daily Values (DV) are based on a 2,000 calorie diet. Your daily values may be higher or lower depending on your calorie needs.</small>		
Ingredients: Cream, Skim Milk, Liquid Sugar, Water, Egg Yolks, Brown Sugar, Milkfat, Peanut Oil, Sugar, Butter, Salt, Carrageenan, Vanilla Extract.		
Price: \$12.72		

Check this box to proceed.

7

If you eat the entire container, how many calories will you eat? *

(1 Point)

8

If you are allowed to eat 60 grams of carbohydrates as a snack, how much ice cream could you have? *

(1 Point)

9

What's the price of the ice-cream? *

10

Your doctor advises you to reduce the amount of saturated fat in your diet. You usually have 42 g of saturated fat each day, which includes one serving of ice cream. If you stop eating ice cream, how many grams of saturated fat would you be consuming each day? *

(1 Point)

11

If you usually eat 2,500 calories in a day, what percentage (%) of your daily value of calories will you be eating if you eat one serving? *

(1 Point)

12

Pretend that you are allergic to the following substances: penicillin, peanuts, latex gloves, and bee stings. Is it safe for you to eat this ice cream? *

(1 Point)

Yes

No

13

If you replied "No" to the previous question, motivate your choice: *

(1 Point)

Main form (2/3)

To complete this form **you will need your assigned tool.**

Your assigned tool is: **\$tool_name**

Please do not seek help from anyone else to complete this form. The goal of this experiment is to assess your ability to use the tool, not to score maximum points at any cost. Your identity will be kept anonymous.

Additional help:

- You are allowed to use a calculator if you would like to.
- We suggest you to use the glossary to better understand the questions.

How to download, setup and use your tool:

Below you can find two download links:

1. **Glossary:** we made this file to make it clearer what certain terms means. You can use it to better understand what we're asking you.
2. **User guide:** this file shows you how to **download, install and setup** the app. It also guides you through all the features that you can use to answer the following questions.

Download links:

- **Glossary:** [\\$glossary_link](#)
- **User guide:** [\\$guide_link](#)

Credentials:

- **Username:** [\\$user](#)
- **Password:** [\\$password](#)

Please open the app and login now before proceeding.

Additional support:

If you have questions or something doesn't work, feel free to contact us at the following email:

14

Please read everything before proceeding, otherwise you could struggle while doing the experiment. *

I read everything!

Food diary on November 28 2021

Following the **user guide**, you can access a **food diary**. That is, for two consecutive weeks **you can see every meal and some related information** (e.g.: nutrients and calories).

Through the app, check **November 28 2021** only and answer the questions to the

15

Which one of the following is true for November 28 2021? *

(1 Point)

- The calorie intake is **too high**.
- The calorie intake **is balanced**.
- The calorie intake **is too low**.
- I don't know.

16

Which one of the following is true for November 28 2021? *

(1 Point)

- The carbohydrates intake is **too high**.
- The carbohydrates intake **is balanced**.
- The carbohydrates intake **is too low**.
- I don't know.

17

Write the single food with most calories on November 28 2021:

(If you're not able to answer just type "unknown" and proceed) *

(1 Point)

18

How many calories does that food contain?

(If you're not able to answer just type 0 and proceed) *

(1 Point)

19

Write the single food with most fat on November 28 2021:

(If you're not able to answer just type "unknown" and proceed)

*

(1 Point)

20

How many grams of fat does that food contain?
(If you're not able to answer type 0 and proceed) *

(1 Point)

21

Describe \$tool_name app logo in your own words: *

Food diary on November 22-28 2021

Following the **user guide**, you can access a **simulated food diary**. That is, for two consecutive weeks **you can see every meal and some related information** (e.g.: nutrients and calories).

Through the app, check the week **November 22-28 2021** only and answer the questions to the best of your knowledge

22

Which one of the following is true for November 22-28 2021? *

(1 Point)

- The calories intake is **too high**.
- The calories intake is **balanced**.
- The calories intake is **too low**.
- I don't know.

23

Which one of the following is true for November 22-28 2021? *

(1 Point)

- The carbohydrates intake is **too high**.
- The carbohydrates intake is **balanced**.
- The carbohydrates intake is **too low**.
- I don't know.

24

Go to the home section of \$tool_name. At the top, you will see a recap of your profile, with a picture. What do you see as the profile picture? *

Food diary on November 15-21 2021 and on November 22-28 2021

Following the **user guide**, you can access a **simulated food diary**. That is, for two consecutive weeks **you can see every meal and some related information** (e.g.: nutrients and calories).

Through the app, check both:
- the week November 15-21 2021
- the week November 22-28 2021

25

Which one of the following is true? *

(1 Point)

- The calorie intake is **better on November 22-28 2021**
- The calorie intake was **better on November 15-21 2021**
- The calories intake **is the same** for both weeks
- I don't know.

26

Which one of the following is true? *

(1 Point)

- The carbohydrates intake is **better on November 22-28 2021**
- The carbohydrates intake was **better on November 15-21 2021**
- The carbohydrates intake **is the same** for both weeks
- I don't know.

27

On November 19 2021, which one of these can you see in "Snacks/Other"? *

- Spaghetti bolognese
- Espresso
- Gin and Tonic

Final feedback (3/3)

Thank you again for your help. In this final form, we ask you to evaluate your overall experience by using your assigned tool.

Please do not give the most positive answer if you don't fully agree with the statement. The goal of this form is to see how good the tool was for you.

28

Please give a score to each statement, based on how much you agree with each one: *

	Disagree	Somewhat disagree	Neither agree or disagree	Somewhat agree	Agree
\$tool_name helped me finding problems in the food diary.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
\$tool_name helped me answer the questions.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Getting the answers with \$tool_name was quick.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
\$tool_name was easy to use and understand.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I think I could improve my diet using \$tool_name .	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

29

Did you use any diet-coaching tool (even \$tool_name itself) before this experiment? *

Yes

No

30

If you chose yes, how often do you use the assigned or similar tool? *

Often

Occasionally

Rarely

Never

Generation of Student Questions for Inquiry-based Learning

Kevin Ros, Maxwell Jong, Chak Ho Chan, ChengXiang Zhai

University of Illinois at Urbana-Champaign

{kjros2, mjong3, chchan2, czhai}@illinois.edu

Abstract

Asking questions during a lecture is a central part of the traditional classroom setting which benefits both students and instructors in many ways. However, no previous work has studied the task of automatically generating student questions based on explicit lecture context. We study the feasibility of automatically generating student questions given the lecture transcript windows where the questions were asked. First, we create a data set of student questions and their corresponding lecture transcript windows. Using this data set, we investigate variants of T5, a sequence-to-sequence generative language model, for a preliminary exploration of this task. Specifically, we compare the effects of training with continuous prefix tuning and pre-training with search engine queries. Question generation evaluation results on two MOOCs show that that pre-training on search engine queries tends to make the generation model more precise whereas continuous prefix tuning offers mixed results.

1 Introduction

It is difficult to understate the importance of asking questions in educational settings. Well-formed questions serve many purposes, including testing student understanding, encouraging exploration of new knowledge, guiding research directions, and developing critical thinking skills (Cotton, 1988). Question-asking also has many benefits for both students and instructors because of its implicit coupling to the context in which questions are asked. For example, instructors can use student questions as implicit feedback to gauge the difficulty of a lecture or to anticipate and update pain points in lecture content. For students, upon hearing a question, they may find it helpful to think about possible answers or to connect the question to their own thought process, thus encouraging inquiry-based learning (Edelson et al., 1999).

However, the benefits of question-asking are much harder to realize in online, asynchronous class settings compared to traditional, in-person class settings. In the latter case, the students and the instructors are co-located and generally, everyone is aware of the current context (i.e., the lecture) and the question. In the former case, the students watch the lectures independently of each other. If a student independently leverages an online search engine to answer a question, then there is no way for their peers and instructors to benefit from the question being asked.

Automatically generating realistic student questions would bring significant benefit to online, asynchronous class settings. For example, instructors could use synthetic student questions to augment lecture videos with additional material. And students could use synthetic student questions to guide studying or to test understanding. Moreover, the synthetic student questions could act as a discussion guide among students and instructors by helping them focus on difficult material.

In this paper, we study how to generate such student questions automatically from given lecture transcript windows. Despite the large amount of previous work regarding question generation (Zhang et al., 2021) (see Section 2 for a detailed review), no previous work has studied our problem setup, as in our case, the answers to the questions may not be available in the lecture transcript content and the questions themselves may not provide enough context to be understood and answered on their own. In virtually all of the data sets used in the existing work, the answers to the questions to be generated are generally assumed to be either directly available or indirectly inferable from the text context. To facilitate the study of this new application scenario of question generation, we create a new data set by collecting and using two MOOC (Massive Open Online Course) transcripts along with 536 questions asked by students

of the MOOCs. Each question includes the corresponding MOOC lecture timestamp window for when the question was asked, thus enabling us to evaluate various context-based question generation approaches.

As an initial investigation of this new task, we focus our exploration on the question generation performance of the generative language model T5 (Raffel et al., 2019) in various settings, leaving a full exploration of different models as future work. Motivated by our small number of training examples, we explore the performance effects of continuous prefix tuning, which has been shown to perform well on natural language generation tasks in low-data settings (Li and Liang, 2021). Additionally, we examine the effects of using docTTTT-query, a T5 model pre-trained with search engine query generation (Nogueira et al., 2019a), on student question generation. Specifically, we investigate the following research questions:

- RQ1:** How does pre-training on search engine query generation affect student question generation performance?
- RQ2:** How does continuous prefix tuning affect student question generation performance?

We find that pre-training on search engine queries tends to make the generation models more precise and that continuous prefix tuning tends to outperform traditional fine-tuning (albeit with mixed significance testing results). Overall, we conclude that it is feasible and promising to use modern machine learning and natural language processing techniques to automatically generate student questions from explicitly-mentioned lecture context in low-data settings.

2 Related Work

Question Generation (Rus et al., 2010; Mazidi and Tarau, 2016) has been extensively studied, initially in the context of generating questions for educational purposes (Mitkov et al., 2006; Kurdi et al., 2020), later with broader application contexts beyond education, such as question answering (Duan et al., 2017) and conversational agents (Wang et al., 2018a).

The survey (Pan et al., 2019) provides a detailed discussion of the major data sets used in recent work on neural question generation and the different levels of questions supported by those data

sets, concluding that the current methods cannot work well for generating deep questions. Virtually all the existing data sets have been generated based on answers in the provided text context (i.e., answer-aware (Zhang et al., 2021)) with perhaps only one exception, which is the LearningQ data set (Chen et al., 2018), where the problem formulation does not include the use of answer when generating a question (i.e., answer-agnostic (Zhang et al., 2021)). Our work is closest to (Chen et al., 2018) in that our formulation of question generation is also answer-agnostic. However, the questions included in the LearningQ data set have been filtered to ensure that the questions included are context-complete. In other words, a question in the LearningQ data set must contain sufficient contextual information on its own to enable other learners to answer the question. Because the structure of our data set explicitly guarantees a reference to the point in the lecture where the question was asked, we can keep questions which don't provide much context themselves (e.g., "Could you be more specific?"). This coupling between lecture and question provides the basis for a new application scenario of question generation where the generated questions are meant to encourage inquiry-based learning (Edelson et al., 1999) for students consuming online lectures. Thus, our data set and approach facilitate a study of how to generate interesting open-ended deep questions using lecture context.

(Ko et al., 2020) collected questions without answers from readers of news articles. They asked study participants to read the first paragraph of various news articles one sentence at a time. If the participant had a question about the sentence, they were instructed to highlight the location of the sentence (e.g., a word or phrase) and write down their question. They collected approximately 19,000 questions and corresponding contexts. However, many of the questions tended to be simpler than the ones collected for our study or by (Chen et al., 2018), and they tended to be answerable by the following sentences in the paragraphs. Moreover, the context sizes selected in (Ko et al., 2020) were smaller than the student-selected lecture windows in our data set.

The survey (Zhang et al., 2021) provides an up-to-date comprehensive review of different lines of work with detailed categorization of the task formulation and comparison of the major approaches

including both rule-based approaches (Lindberg et al., 2013) and modern neural network-based approaches (Du et al., 2017; Duan et al., 2017; Lewis et al., 2019). In our work, because the data set has a small number of training examples, while the question structures can be quite complex, we focused on exploring the use of general pre-trained language models (T5) and prefix tuning / pre-training on search engine queries to address the technical challenges.

More generally, the notion of question generation has been studied from the perspective of information retrieval (Nogueira et al., 2019a,b). Here, the authors expanded documents with the queries for which the documents would be relevant. For a given search query, a document’s relevance score was computed using both the document’s content and its respective generated queries. We leverage their fine-tuned query generation model, docTTTT-Tquery, as the basis for answering **RQ1**.

Previous work has found that learners engage heavily with in-video quizzes (Kovacs, 2016). However, such quizzes are usually designed manually. The methods that we explore can be potentially used to automatically generate in-video questions to enhance learner engagement. Automatic generation of quiz questions for testing learners’ knowledge has also been attempted. For example, Wang et al. introduced QG-Net, a recurrent neural network-based model that can generate quiz questions from educational content (Wang et al., 2018b). Although such questions were also generated based on educational content, their answers were generally available in the educational content from which the questions are generated; in contrast, in our work, the answers to those questions generated are generally not available directly in the educational content.

3 The Lecture-Question Data Set

Because our exact problem setup has not been studied before, there does not exist any data set that we can use for our experiments. Thus, we created a data set from student questions previously submitted to two MOOCs available on Coursera, titled *Text Retrieval and Search Engines* and *Text Mining and Analytics*. The students were enrolled in a class which used the two MOOCs as the major lectures. There are a total of 90 lecture across the MOOCs, and each lecture contains a complete transcript of the audible instruction, with frequent and

regular timestamps. Students were asked to submit any questions that they had about the MOOC lecture content, and to include a reference to the lecture name and timestamps where the question occurred. The question submission template was as follows:

<Lecture name, start time, end time, question>

Both of the MOOC transcripts and the submitted student questions are in English.

3.1 Preprocessing and Data Availability

To clean the data, we filtered out all questions with malformed annotations or missing timestamps. Next, we attempted to map each question to the respective lecture transcript window. The overall cleaning process resulted in a data set of 536 (lecture window text, question text) pairs. This is the data set that we used to quantify the performance of question generation. The original and filtered data sets, as well as the complete MOOC transcripts, are available on GitHub.¹ We obtained IRB approval and permission from the MOOC author to release the anonymous data. Note that we removed any student-identifiable information from the data set.

3.2 Basic Properties of the Student Questions

This section describes the characteristics of the student questions from our filtered data set. For the 536 questions, the mean number of words per question is 18, and the median number of words is 15. A similar skew is also present in the corresponding lecture windows. The mean number of words in each lecture window is 210, and the median number of words is 132. Moreover, the mean number of seconds in each window is 92, whereas the median number of seconds is 60.

Unigrams	Trigrams
is (264)	what is the (65)
how (188)	how do we (30)
what (181)	the meaning of (13)
does (107)	the difference between (12)
why (104)	why do we (11)
are (103)	is it possible (9)

Table 1: A list of some of the most frequent unigrams and trigrams in student questions. The number in parentheses indicates the occurrence frequency.

¹<https://github.com/kevinros/INLG2022StudentQuestions>

Question Examples
What is the point of compression? Will the access times really be that impactful to the overall indexing?
Are the doc-ids sorted with the term-ids in the "local" sort?
Can we get more examples of using gamma-code?
How does the gamma-code intergar compression method work? I did not understand the example from the video
I'm still very confused how integer compression actually reduces size of storage since some of the examples make it seem like you're using more bits than before on some inputs

Table 2: A few example questions from the lecture-question data set.

Table 1 depicts a list of some of the most common unigrams and trigrams present in student questions. Interestingly, many questions are concerned with the meaning or difference of the referenced content. The most common interrogative words are "how", "what", and "why". A few examples of questions from our data set are presented in Table 2. Note that there is significant noise in the data set: there are misspellings (e.g., "interger" in the third question), multiple questions submitted as one question, and general expressions of confusion instead of questions. Although we expect that our models would perform better if we removed noise through additional preprocessing, we felt that it was important to remain as close as possible to the original data to reflect a scaled learning scenario where manual preprocessing is impractical.

There are some clear limitations of our collected lecture-question data set. The size of the data set and the MOOC topic similarity make it difficult to know if our findings generalize to different data sets. Also, the lecture transcripts and questions are written in English, which certainly limit applicability. However, given the overall lack of data for this problem setting, we hope that our data set and methods can offer a starting point for future researchers and educators to extend student question generation into more general settings.

4 Methods for Question Generation

4.1 Problem Formulation

We now briefly formalize the proposed task of student question generation from lecture transcripts. Our data set consists of (L_i, Q_i) pairs. In each pair, lecture window $L_i = (w_j)_{j=1}^{n_i}$ is a sequence of n_i word tokens. The start and end positions of lecture window L_i are determined by the start and end timestamps submitted in the question Q_i .

Question $Q_i = (q_k)_{k=1}^{m_i}$ is a sequence of m_i word tokens. We aim to generate question Q_i given its corresponding lecture window L_i .

Formally, we model question generation as a sequence-to-sequence language generation task. Let model M be a sequence-to-sequence language model initialized with trainable parameters ϕ . Our goal is to maximize the probability of $M_\phi(Q_i|L_i)$. In other words, the input to model M is lecture window L_i and the desired output is the corresponding student question Q_i .

4.2 RQ1: T5 and docTTTTTquery

For the sequence-to-sequence language model architecture, we use T5 (Raffel et al., 2019), which is based on the standard encoder-decoder transformer architecture (Vaswani et al., 2017). We choose T5 due to its state-of-the-art performance in the text summarization task, which closely resembled our question generation problem formulation (i.e., "summarizing" the lecture window as a question). The idea behind T5's original implementation was to improve performance by having a single model learn many different tasks (translation, summarization, classification, etc.) as sequence-to-sequence text tasks via discrete fixed prompt instructions (Liu et al., 2021). Specifically, each training example began with a pre-defined discrete prompt (e.g., "translate English to German:") which served the purpose of instructing the model to handle the input data according to the task described in the prompt.

We test two existing instantiations of T5, namely t5-base and docTTTTTquery (Nogueira et al., 2019a). The former was trained in accordance with the original T5 paper and the latter is a version of t5-base fine-tuned on query generation given relevant passages using the MS-MARCO data set (Nguyen

et al., 2016). To answer **RQ1**, we fine-tune both models to determine if a model pre-trained to generate search engine queries offers any performance benefits on our student question generation task. For completeness, we also include the performance of the base docTTTTTquery model that is not fine-tuned on our data set.

4.3 RQ2: Continuous Prefix Tuning

Continuous prefix tuning, proposed by (Li and Liang, 2021), is a method for fine-tuning large generative models that has been shown to perform well in low-data scenarios. We are interested in studying the effects of continuous prefix tuning for generating student questions. Thus, we adapt Li and Liang’s approach to T5 and measure the performance on our collected lecture-question data set. We now provide a formal overview of their continuous prefix tuning applied to question generation.

For the continuous prefix tuning setting, language model M is still initialized using pre-trained parameters ϕ . Consider a continuous prefix $p \in \mathbb{R}^{d \times n}$. Here, d is the input embedding dimension of model M and $n \in \mathbb{N}_{\geq 0}$ is the chosen length of the prefix, which must be strictly less than the maximum sequence length of M . The input to the language model then becomes $L'_i = [p; L_i]$. The goal is to maximize $M_{\phi,p}(Q_i|L'_i)$ where parameters ϕ are fixed and prefix p is free. In other words, we freeze the original parameters of the language model and aim to learn the values of p which best help M generate Q_i . Note that p is the same across all training, validation, and testing pairs.

For our implementation, we also follow the continuous prefix tuning reparameterization approach of (Li and Liang, 2021), which they found to increase training stability. As noted earlier, the prefix p is a continuous matrix with values determined by the fine-tuning process on the training data. To determine the values of p , we fix hyperparameter $n' \in \mathbb{N}_{>0}, n' \leq n$ and randomly initialize $p' \in \mathbb{R}^{d \times n'}$. Then, we define p as the output of a two-layer neural network N parameterized by ψ . In other words, we compute prefix p as $p = N_\psi(p')$, where parameters ψ are learned during training. For our experiments, N_ψ is a fully-connected two layer neural network with hidden dimension h .

To answer **RQ2**, we fine-tune t5-base and docTTTTTquery with the continuous prefix modification and compare the resulting performance metrics to the traditionally fine-tuned models.

4.4 Evaluation

We randomly split the 90 MOOC lectures into 85 training lectures and 5 testing lectures. We split on the lecture level to avoid the possibility of the model seeing overlapping windows during training and testing. This split results in 483 questions in the training set and 53 questions in the testing set. Then, for three iterations, we randomly hold out 10 lectures from the 85 training lectures as a validation set. The validation lecture sets are disjoint across all iterations, and each model is trained and validated on the same splits. The hyperparameters for each model are selected based on the highest averaged ROUGE-1 F_1 score over the validation splits. After selecting the hyperparameters, we retrain the model on the entire 85 lecture training set. To measure the performance of our trained models, we report the single-run precision, recall, and F_1 score for the ROUGE-1, ROUGE-2, and ROUGE-L measurements averaged across all generated and ground-truth questions in the testing set. All ROUGE scores are computed using the default settings of the rouge-score python package.² Regarding hyperparameter selection, the number of epochs ranges from [1, 10] and the learning rate ranges from {1e-6, 1e-5, 1e-4}.

5 Question Generation Results

Table 3 contains the single-run ROUGE-1, ROUGE-2, and ROUGE-L scores on the test set for each best-performing model on the validation set. The first column lists the name of each model. "FT" refers to traditional fine-tuning and "Prefix" refers to continuous prefix tuning. For the run labeled "docTTTTTquery", we evaluate the model on the test set without any training. There is no corresponding "t5-base" run because the original model was not trained to generate questions. The second column labeled "R" is the recall, the third column labeled "P" is the precision, and the fourth column labeled " F_1 " is the F_1 score.

5.1 Hyperparameters

The final hyperparameter selections for each model are reported in Table 4. Each validation run takes approximately one hour on a single Nvidia GeForce 1070x GPU with a batch size of one. Note that the number of parameters are essentially the same

²<https://pypi.org/project/rouge-score/>
Rouge-score is released under Apache License 2.0, which permits research use.

Model	R	P	F ₁
	ROUGE-1 (%)		
t5-base FT	20.06	14.47	14.82
t5-base Prefix	20.13	21.56	18.63
docTTTTTquery	14.41	25.17	16.83
docTTTTTquery FT	15.70	23.34	17.45
docTTTTTquery Prefix	17.19	24.00	18.74
	ROUGE-2 (%)		
t5-base FT	1.697	1.656	1.502
t5-base Prefix	3.267	3.391	3.043
docTTTTTquery	3.237	4.596	3.358
docTTTTTquery FT	4.011	4.730	3.903
docTTTTTquery Prefix	4.790	6.247	5.010
	ROUGE-L (%)		
t5-base FT	15.82	11.57	11.77
t5-base Prefix	16.89	17.65	15.47
docTTTTTquery	13.17	22.32	15.18
docTTTTTquery FT	14.34	20.64	15.76
docTTTTTquery Prefix	15.47	21.00	16.73

Table 3: The recall, precision, and F₁ scores for ROUGE-1, ROUGE-2, and ROUGE-L measurements for the question generation approaches on the test set. "FT" refers to traditional fine-tuning and "Prefix" refers to continuous prefix tuning.

across both models (220M (Raffel et al., 2019)), as docTTTTTquery is a fine-tuned version of t5-base. Additionally, the prefix reparameterization parameters can be dropped once the model is trained. Preliminary experiments indicated that larger prefixes tended to perform better, so we fix the prefix length (n , in Table 4) to be sufficiently large. The random seed is set to 42 across all runs. All other hyperparameters are set to the default settings.

5.2 Answering RQ1

To answer **RQ1**, we compare the performance of the t5-base models with the performance of the docTTTTTquery models. Beginning with the FT models for both cases, we find that docTTTTTquery FT has lower ROUGE-1 and ROUGE-L recall scores than t5-base FT but higher ROUGE-1 and ROUGE-L precision scores. Additionally, docTTTTTquery FT has a higher recall and precision for ROUGE-2. In all three ROUGE cases, docTTTTTquery FT has a higher F₁ score than t5-base FT. There is also a similar trend for the Prefix models. Namely, docTTTTTquery Prefix has lower ROUGE-1 and ROUGE-L recall scores than t5-base Prefix, but higher precision and F₁ scores. Moreover, docTTTTTquery Prefix has a higher re-

call, precision, and F₁ score for the ROUGE-2 measurement. Based on the averages of the ROUGE scores, we see that the docTTTTTquery models are generally more precise than the t5-base models. To test the statistical significance of the precision improvement, we performed a one-tailed Wilcoxon signed-rank test comparing the precision of each t5-base run to its respective docTTTTTquery run. We selected the Wilcoxon signed-rank test because of its non-parametric property, as the distributions of precision appear to be non-normal. All runs were significant at $p = 0.05$, except for the ROUGE-1 precision between the t5-base Prefix model and the docTTTTTquery Prefix model ($p = 0.077$), and the ROUGE-L precision between the t5-base Prefix model and the docTTTTTquery Prefix model ($p = 0.080$). In conclusion, pre-training on search engine query generation appears to offer clear benefit in increasing the precision, though the benefit appears to be more for traditional fine-tuning.

5.3 Answering RQ2

To answer **RQ2**, we compare the performance of the FT model variants with the Prefix model variants. Beginning with the t5-base models, we find that the runs have similar recall scores for ROUGE-1, whereas the t5-base Prefix has higher recall scores for ROUGE-2 and ROUGE-L. The precision and F₁ scores for all ROUGE measurements are higher for t5-base Prefix. For the docTTTTTquery models, we find that docTTTTTquery Prefix has the highest recall across all three ROUGE measurements. There is no clear trend for the precision. However, the increases in recall are enough for docTTTTTquery Prefix to have the highest F₁ scores for all three ROUGE measurements. Similar to the previous research question, we performed a one-tailed Wilcoxon signed-rank test comparing the F₁ scores of each Prefix model to its respective FT model, in order to test for improvement. Only the ROUGE-1 and ROUGE-L scores between the t5-base FT model and the t5-base Prefix model were significant at $p = 0.05$. Note that the ROUGE-2 score comparison between the docTTTTTquery FT model and the docTTTTTquery Prefix model had $p = 0.055$. From these results, there seems to be marginal benefit for using continuous prefix tuning in a low-data setting to generate student questions.

Model	Learning Rate	Epochs	n'	h	n	dropout	hidden activation
t5-base FT	1e-5	8	-	-	-	-	-
t5-base Prefix	1e-5	7	20	800	100	0.5	tanh
docTTTTTquery	-	-	-	-	-	-	-
docTTTTTquery FT	1e-5	7	-	-	-	-	-
docTTTTTquery Prefix	1e-4	7	20	800	100	0.5	tanh

Table 4: Best-found hyperparameters for each trained model on the validation set. The last five columns correspond to the prefix reparameterization hyperparameters described in Section 4.3.

5.4 Qualitative Analysis

Because our testing set is small, quantitative analysis and significance testing may not offer a clear picture into the qualitative differences between the models. Therefore, we perform a brief qualitative comparison among the ground truth questions and the generated questions for each model. A few question examples are presented in Table 5, and the corresponding lecture windows are presented in Table 6. These examples were hand-selected to demonstrate some interesting characteristics of the question generation models.

The selected examples presented in Table 5 offer some possible explanations for the variations in ROUGE scores from Table 3. Notably, the docTTTTTquery models typically generate shorter questions than the t5-base models. The average number of words generated per question by the t5-base FT and t5-base Prefix model were 24 and 15 respectively, whereas the docTTTTTquery, docTTTTTquery FT, and docTTTTTquery Prefix averages were 8, 10, and 10, respectively. Additionally, t5-base FT sometimes generates multiple subquestions for a single ground truth. This may explain why the docTTTTTquery models generally have higher precision scores but lower recall scores. We believe that the docTTTTTquery models generate shorter questions because search engine queries tend to be much shorter than our collected student questions (Craswell et al., 2020). Moreover, the similarity of generated questions (e.g., between the docTTTTTquery models in the second example) may help explain the non-significant score results. It is also important to note that non-significant differences in ROUGE score may not imply non-significant differences in question meaning or quality. In the first example of Table 5, the docTTTTTquery FT generated question and the docTTTTTquery Prefix generated question only differ by a word ("delta" versus "gamma"). Despite this small difference, the questions have completely different

meaning and answers.

Another interesting observation is the apparent inability of the models to capture mathematical expressions. For example, in the second question group, all models miss "(1+logx)" and "x-2(logx)". One possible explanation is that the transcript does not fully capture these equations, which can be observed in the second row of Table 6. This might be addressable by incorporating multi-modal contexts (e.g., lecture transcripts *and* lecture slides), which is an interesting direction for future work.

We also observe cases where there is a mismatch between the topics present in the ground truth and the topics present in the generated questions. This seems to indicate that a more controlled generation approach may be necessary.

5.5 Limitations

There are a few limitations of our generation experiments. First, due to the difficulty in collecting the data for our study, the data set that we have managed to create at this point is small. While the data set was already useful to allow us to investigate multiple variants of T5-based models to make some preliminary conclusions, it is unclear to what extent we can generalize those conclusions. Thus, in the future, it is important to further increase the size of the data set, which we are planning to do by continuously collecting the questions from students in future classes. With larger data sets, we can also more thoroughly explore the cutting-edge models for this task. Second, our notion of question "ground truth" may not be the best way to capture students' questions. This is because there can be many equally-valid questions for a given lecture window, even if they are different from one another. Third, the ROUGE scores may not effectively capture the high-level semantics of a question. That is, small changes in question word choice or structure can result in similar ROUGE scores while drastically altering the meaning.

Model	Question
Ground Truth	Does is the delta-code use gamma-code twice recursively?
t5-base FT	What is the difference between delta coding and delta coding? Is it possible to use delta coding for inverted index distribution?
t5-base Prefix	What is the difference between delta and gamma?
docTTTTTquery	what is gamma coding
docTTTTTquery FT	what is the difference between delta and delta coding?
docTTTTTquery Prefix	what is the difference between delta and gamma coding?
Ground Truth	What is the reasoning for making the first $(1+\log x)$ unary and the $x-2(\log x)$ uniform? The method for encoding seem random to me.
t5-base FT	What is the problem? Is there a way to use binary code to calculate the remaining part of the value of x ?
t5-base Prefix	what is the difference between unary code and binary code?
docTTTTTquery	what is the difference between unary and binary code
docTTTTTquery FT	what is the difference between unary and binary code?
docTTTTTquery Prefix	what is the difference between unary and binary code?
Ground Truth	What is the risk associated with discarding documents that are potentially relevant?
t5-base FT	is there any way to determine the most likely relevance in the documents?
t5-base Prefix	What is the purpose of having a diverse set of ranking methods?
docTTTTTquery	what is diverse set of ranking methods
docTTTTTquery FT	what is the difference between a diverse set of ranking methods and an unjudged pool of documents?
docTTTTTquery Prefix	what is meant by having a pool of relevant documents that aren't being ranked?

Table 5: A few examples of the ground truth question compared to each model’s generated question.

Nevertheless, the scores presented in Table 3 and the examples presented in Table 5 indicate that it is possible to generate meaningful student questions from lecture content with low amounts of data. Moreover, we find benefits in the use of continuous prefix tuning and in the use of search engine queries to fine-tune pre-trained language models for question generation. Overall, we hope that our results can help guide future researchers for designing student question generation models in similar low-data settings.

6 Conclusion and Future Work

In this paper, we studied a new application scenario of question generation, where the goal is to generate interesting questions that can promote inquiry-based learning for students watching online lecture videos. The task is different from many existing question generation tasks in that the answers to the questions may not be available in the text context used to generate a question. We created and released a new data set for studying this problem. We also studied how to use various T5 models to

solve the problem effectively. Experimental results showed that the task is challenging, but continuous prefix tuning and pre-training on search engine queries show promise in the direction of generating coherent and relevant questions in spite of limited training data. Moreover, the ability to use search engine queries as pre-training data hints at the scalability of precise student question generation due to the wide availability of queries.

Full exploration of the potential of the proposed methods and further evaluation of the benefits of the generated questions for real learners are important directions for future work. One particularly promising albeit difficult area for future work is to consider a more fine-grained question generation approach by conditioning the generation model not only on the lecture context but also the student context (i.e., a student’s background knowledge of a subject). Additionally, generation could be framed in the context of multiple lecture locations at once instead of a single window. Another possible direction is to investigate methods for using language models to generate student questions about mathe-

Question	Lecture Window
Does is the delta-code use gamma-code twice recursively?	except that you replace the unary prefix with the gamma code. So that's even less conservative than gamma code, in terms of avoiding the small integers. So that means it's okay if you occasionally see a large number. It's, it's, you know, it's okay with delta code. It's also fine with gamma code. It's really a big loss for unary code, and they are all operating, ...
What is the reasoning for making the first $(1+\log x)$ unary and the $x-2(\log x)$ uniform? The method for encoding seem random to me.	this is basically the same uniform code and binary code are the same. And we're going to use this code to code the remaining part of the value of x . And this is basically, precisely, x minus 1, 2 to the flow of \log of x . So the unary code or basically code with a flow of \log of x , well, I added one there, and here. But the remaining part will, we using uniform code to actually code the difference between the x and
What is the risk associated with discarding documents that are potentially relevant?	We would first choose a diverse set of ranking methods, these are types of retrieval systems. And we hope these methods can help us nominate likely relevance in the documents. So the goal is to pick out the relevant documents.. It means we are to make judgements on relevant documents because those are the most useful documents from the users perspective...

Table 6: The lecture windows corresponding to the questions presented in Table 5.

mathematical formulas. Finally, it would be interesting to further explore alternative controlled methods for question generation in low-data settings, such as few-shot approaches or simpler, rule-based approaches.

Our proposed task of generating questions from indicated lecture content is inherently applied in nature, as it is centered around learners and instructors in an educational setting. Thus, future work should also consider the direct utility of learners and instructors as a future measure of model effectiveness. And as we discussed in Section 5.5, the "ground truth" question for a given context may not be consistent across individuals. Or, for a given individual, different questions may have different dimensions of utility. This leads to an interesting direction of exploring the types of questions that individuals find useful in various contexts.

In a more general sense, our problem setting could be cast in an outward direction by examining the reasons behind why learners ask questions or by examining the linguistic structures and characteristics of the asked questions. Better understandings of these directions may help drive more efficient or simpler model architectures, training procedures, and evaluation metrics.

With the growth of online education, particularly in the context of MOOCs, both instructors and students will find it valuable to be able to better under-

stand, contemplate, and anticipate question-based interactions with course material. We thus hope our preliminary exploration provides a basis for future work on question generation in this application context, eventually creating natural language generation techniques that can be deployed on an online learning platform to automatically generate relevant questions to many online lectures and support inquiry-based learning for many online students.

7 Acknowledgements

This work is supported in part by the National Science Foundation under Grant No. 1801652. We would also like to thank Dr. Heng Ji for her insightful comments.

8 Ethical Impact

As with any natural language generation approach that leverages large pre-trained models, there is the possibility of generating biased or offensive content. Careful consideration is needed to apply these findings to live scenarios, as there likely are many untested or unexpected behaviors of the underlying language models.

References

Guanliang Chen, Jie Yang, Claudia Hauff, and Geert-Jan Houben. 2018. LearningQ: a large-scale dataset

- for educational question generation. In *Twelfth International AAI Conference on Web and Social Media*.
- Kathleen Cotton. 1988. Classroom questioning. *School improvement research series*, 5:1–22.
- Nick Craswell, Daniel Campos, Bhaskar Mitra, Emine Yilmaz, and Bodo Billerbeck. 2020. ORCAS: 20 million clicked query-document pairs for analyzing search. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2983–2989.
- Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. *arXiv preprint arXiv:1705.00106*.
- Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874.
- Daniel C Edelson, Douglas N Gordin, and Roy D Pea. 1999. Addressing the challenges of inquiry-based learning through technology and curriculum design. *Journal of the learning sciences*, 8(3-4):391–450.
- Wei-Jen Ko, Te-Yuan Chen, Yiyang Huang, Greg Durrett, and Junyi Jessy Li. 2020. Inquisitive question generation for high level text comprehension. *arXiv preprint arXiv:2010.01657*.
- Geza Kovacs. 2016. Effects of in-video quizzes on mooc lecture viewing. In *Proceedings of the third (2016) ACM conference on Learning@ Scale*, pages 31–40.
- Ghader Kurdi, Jared Leo, Bijan Parsia, Uli Sattler, and Salam Al-Emari. 2020. A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education*, 30(1):121–204.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.
- Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.
- David Lindberg, Fred Popowich, John Nesbit, and Phil Winne. 2013. Generating natural language questions to support learning on-line. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 105–114.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.
- Karen Mazidi and Paul Tarau. 2016. Infusing NLU into automatic question generation. In *Proceedings of the 9th International Natural Language Generation conference*, pages 51–60.
- Ruslan Mitkov, Ha Le An, and Nikiforos Karamanis. 2006. A computer-aided environment for generating multiple-choice test items. *Natural language engineering*, 12(2):177–194.
- Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.
- Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. 2019a. From doc2query to docTTTTTquery. *Online preprint*.
- Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019b. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*.
- Liangming Pan, Wenqiang Lei, Tat-Seng Chua, and Min-Yen Kan. 2019. Recent advances in neural question generation. *arXiv preprint arXiv:1905.08949*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.
- Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Cristian Moldovan. 2010. The first question generation shared task evaluation challenge. In *Proceedings of the Sixth International Natural Language Generation Conference (INLG 2010)*, pages 105–114.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.
- Yansen Wang, Chenyi Liu, Minlie Huang, and Liqiang Nie. 2018a. Learning to ask questions in open-domain conversational systems with typed decoders. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2193–2203, Melbourne, Australia. Association for Computational Linguistics.
- Zichao Wang, Andrew S Lan, Weili Nie, Andrew E Waters, Phillip J Grimaldi, and Richard G Baraniuk. 2018b. Qg-net: a data-driven question generation model for educational content. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*, pages 1–10.
- Ruqing Zhang, Jiafeng Guo, Lu Chen, Yixing Fan, and Xueqi Cheng. 2021. A review on question generation from natural language text. *ACM Transactions on Information Systems (TOIS)*, 40(1):1–43.

Keyword Provision Question Generation for Facilitating Educational Reading Comprehension Preparation

Ying-Hong Chan

Ho-Lam Chung

Yao-Chung Fan

Department of Computer Science and Engineering
National Chung Hsing University,
Taichung, Taiwan

Abstract

Question Generation (QG) receives increasing research attention in the NLP community. One QG motivation is to facilitate the preparation of educational reading practice and assessments. While significant advancement of QG techniques was reported, we find current QG techniques are short in terms of *controllability* and *question difficulty* for educational applications. This paper reports our studies toward the two issues. First, we report a state-of-the-art exam-like QG model by advancing the current best model from 11.96 to 20.19 (in terms of BLEU 4 score). Second, we propose a QG model that allows users to provide keywords for guiding QG direction. Human evaluation and case studies are conducted to demonstrate the feasibility of controlling question generation direction.

1 Introduction

Question generation (QG), taking a passage and an answer phrase as input and generating a context-related question as output, has received interest in recent years (Zhou et al., 2017; Zhao et al., 2018; Du et al., 2017; Chan and Fan, 2019; Dong et al., 2019; Bao et al., 2020). One motivation for developing QG is to facilitate educators in the preparation of reading comprehension assessments.

While significant QG quality was reported, we find two limitations for integrating the current QG models into educational usage scenarios.

First, the current QG model suffers from the model controllability concern. In Table 1, we show an example with a passage, an answer, and two questions (Q_1 and Q_2). The model controllability concern lies in that we have no way to control the QG direction with the model (Chan and Fan, 2019; Dong et al., 2019; Bao et al., 2020).

We note that both questions have the same answer (i.e., *Christopher Hirata*), while the models are designed to take a context and an answer span as input for QG. Thus, there are no way to control which question to generate.

Context	At the age of 12, Christopher Hirata already worked on college-level courses, around the time most of us were just in the 7th grade. At the age of 13, this gifted kid became the youngest American to have ever won the gold medal in the International Physics Olympiad. At the age of 16, he was already working with NASA on its project to conquer planet mars. After he was awarded the Ph.D. at Princeton University, he went back to California institute of technology. The next person with a very high IQ is Albert Einstein. With an IQ between 160 and 190, Albert Einstein is the genius behind the theory of relativity, which has had a great impact on the world of science.
Answer	Christopher Hirata
Q_1	Who once worked on the project to conquer planet mars?
Q_2	Who was the youngest American to have ever won the gold medal in the International Physics Olympiad?

Table 1: An Example for QG Model Controllability Concern: With the existing QG settings, we have no way to control which question to generate.

Second, questions generated by existing QG models are too simple (in terms of difficulty) for advanced educational reading practice assessment. Current data-driven QG models are trained with factoid QA datasets (e.g., SQuAD (Rajpurkar et al., 2016) or NewsQA (Trischler et al., 2016)), and therefore generate factoid questions, which are too simple for advanced reading practice assessment.

In this paper, we report our results toward the two limitations. First, we propose a new QG setting variant for the controllability issue, which allows users to guide the QG direction by indicating keywords (Please see Section 2). Our design, KPQG (Keyword Provision Question Generation) model, successfully enables QG controllability. Experiments are conducted using benchmark datasets to show the quality of our KPQG model. We also conduct quantitative studies to examine the controllability and feasibility of the generation in various aspects

For the issue of generating too simple questions, we investigate training QG models with exam-like datasets (e.g., RACE (Lai et al., 2017)). We investigate the employment of pre-trained language

models (LM) for exam-like QG. Our experiment results show that the LM employment significantly advances the state-of-the-art result reported by (Jia et al., 2020) from 11.96 to 20.19 (in terms of BLEU 4 score).

2 Methodology

In Subsection 2.1, we first review the existing LM architectures for QG, which are basic building blocks for QG based on LM. In Subsection 2.2, we present Keyword Provision Question Generation (KPQG) scheme for guiding QG generation.

Problem Formulation In this paper, we consider a QG setting that takes (1) a context passage, (2) answer phrase, and (3) *a set of keywords* as input and generate a question contains the keywords as output. Note that the existing QG setting takes only (1) a context passage and (2) answer phrase as input. The idea is to design QG to take additional keywords for question generation. We refer readers to the example illustrated in Figure 1.

2.1 QG Architecture

In this paper, we explore two QG architecture.

Masked-LM Generation The QG model by Masked-LM Generation works as follows. A Masked-LM QG generation model $\mathbb{M}()$ takes a context paragraph C , answer A , and the previous generated tokens q_1, \dots, q_{i-1} and as input and output a target token q_i in an auto-regressive manner, where $[S]$ and $[M]$ are the sep and masked special tokens in pre-trained language models.

$$\begin{aligned} \mathbb{M}(C [S] A [M]) &\rightarrow q_1, \\ \mathbb{M}(C [S] A [S] q_1 [M]) &\rightarrow q_2, \\ \mathbb{M}(C [S] A [S] q_1, q_2 [M]) &\rightarrow q_3, \\ &\dots \end{aligned}$$

Seq2Seq Generation A seq2seq model $\mathbb{M}()$ for QG takes a context paragraph C and an answer A as input and predicting a sequence of question tokens $\{q_1, q_2 \dots q_{|Q|}\}$ as output. Specifically, we have

$$\mathbb{M}(C [S] A) \rightarrow q_1, q_2, \dots, q_{|Q|}$$

2.2 Key Provision Question Generation

Inference Our KPQG model extends the Masked-LM Generation as follows. For a given keyword sequence $[k_1, \dots, k_i]$, a context C and an answer phrase A , the input sequence X to a LM model

is to interleavely place $[M]$ tokens between the keyword sequence as follows.

$$X = [C [S] A [S] [M_1] k_1 [M_2] \dots [M_i] k_i]$$

We leverage Masked-LM generation to predict the $[M]$ tokens. After the prediction, we recursively insert and predict the $[M]$ tokens in the same manner. At each iteration, we align the input sequence by inserting $[M]$ before and after all given/generated tokens. The iteration continues till all masked tokens become $[S]$.

As a concrete example, please refer to the example shown in Figure 1 and Table 2. Two keywords (project and mars) are given in this example. At Iteration 0, we have three inserted $[M]$ tokens, and the predicted results are “Who”, “planet”, and “?”. And, at Iteration 1, we set the input sequence X_1 by inserting $[M]$ before and after all given/generated tokens. The $[M]$ placement and prediction loops until all $[M]$ s becomes $[S]$.

Training to Generate Important Token First

The KPQG is trained to predict a masked token before/after the input/generated keyword tokens. Under this goal, the challenge lies in which tokens should be masked for model training.

We explore the idea of learning to predict important words by employing a QA model (e.g., SQuAD) to assess the importance of tokens. Our idea is that if masking some token q_i from a question sentence $[q_1, \dots, q_{|Q|}]$ leads to a decreased QA model performance, then q_i shall be an important one. Therefore, for a given Q , we iteratively replace all tokens in Q with a $[PAD]$ token in a one-at-a-time manner.

For example, for the question “how is the weather today?”, we have the following *padded* question sentences.

- $[PAD]$ is the weather today?
- how $[PAD]$ the weather today?
- how is $[PAD]$ weather today?
- how is the $[PAD]$ today?
- how is the weather $[PAD]$?
- how is the weather today $[PAD]$

We then post the sentences to a QA model for answer prediction, and estimate the importance of a keyword through the model’s confidence in answer prediction.

KPQG Inference Example

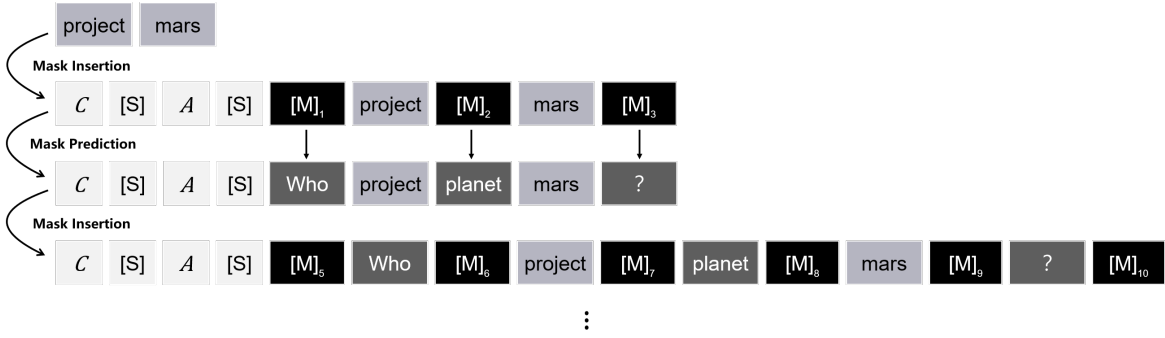


Figure 1: KPQG Mask Insertion and Prediction

After the token importance assessment, we generate training data for KPQG based on the token importance by masking important word first. In Table 3, we show an example. Assume that the importance of a question sentence $[q_1, \dots, q_9]$ is $[q_4, q_6, q_2, q_5, q_3, q_1, q_9, q_7, q_8]$ (from high to low).

As shown in Table 3, six training instances are generated. The first training instance aims to instruct the KPQG model to predict the most important word (i.e., q_4) based on only C and A . That is, the label of the $[M]$ token is set to q_4 .

$$\mathbb{M}(C [S] A [S] [M]) \rightarrow q_4$$

Likewise, the second training instance is set to predict q_2 and q_6 as follows.

$$\mathbb{M}(C [S] A [S] [M] q_4 [M]) \rightarrow q_2, q_6$$

Please refer to the complete training instances in Table 3.

3 Performance Evaluation

3.1 Educational QG Comparison

In this subsection, we report our results on the employment of pre-trained language models (PLM) for educational QG.

We evaluate the results on EQG-RACE (Jia et al., 2020) dataset. Table 4 summarizes statistics for the datasets. We implement the following QG models.

- Masked-LM QG architecture with BERT (Devlin et al., 2018)
- Masked-LM QG architecture with RoBERTa (Liu et al., 2019)
- Masked-LM QG architecture with DeBERTa (He et al., 2020)

- Seq2Seq QG architecture with BART (Lewis et al., 2019)

Table 5 shows the evaluation results on test data. We also list the state-of-the-art result reported by (Jia et al., 2020). We see that the PLM employment significantly improves the performance of educational QG. Among them, DeBERTa-QG advances the SOTA result from 11.96 to 20.19 (in terms of BLEU 4 score).

3.2 KPQG Performance Evaluation

3.2.1 Implementation Details

We use the DeBERTa_{base} (He et al., 2020) model for KPQG training. The KPQG model is trained by four TITAN V100 GPUs with 10 epochs for 16 hours. In addition, for the QA model for assessing token importance for training data preparation, we use the RACE QA model from (Wolf et al., 2020). This model has an accuracy of 84.9% on the RACE dataset.

3.2.2 Human Evaluation

We use human evaluation to validate the quality of the KPQG model because the premise of the KPQG model allows users to guide the QG direction by indicating keywords expected to be included in the generation result. 300 context paragraphs and the corresponding answers were randomly selected from the test set of EQG-RACE data (Jia et al., 2020). We invited 30 evaluators. Each one was given 10 contextual paragraphs and asked to use the KPQG model to provide keywords to generate questions. The evaluator is asked to compare the difference between QG and KPQG and score $[0,1,2]$ on the Likert scale based on the following three metrics:

	M_i	Prediction for [M]
iter0	C [S] A [S] [M] project [M] mars [M]	Who, planet, ?
iter1	C [S] A [S] [M] Who [M] project [M] planet [M] mars [M] ? [M]	[S], worked, to, [S], [S], [S]
iter2	C [S] A [S] Who [M] worked [M] project [M] to [M] planet mars ?	once, the, [S], conquer
iter3	C [S] A [S] Who [M] once [M] worked [M] the [M] project to [M] conquer [M] planet mars ?	[S], [S], on, [S], [S], [S]
iter4	C [S] A [S] Who once worked [M] on [M] the project to conquer planet mars ?	[S], [S]
end	Who once worked on the project to conquer planet mars ?	

Table 2: KPQG Inference Example

	X_i	Labels for [M]
i=0	C [S] A [S] [M]	q_4
i=1	C [S] A [S] [M] q_4 [M]	q_2 q_6
i=2	C [S] A [S] [M] q_2 , [M] q_4 [M] q_6 [M]	q_1 q_3 q_5 q_9
i=3	C [S] A [S] [M] q_1 [M] q_2 [M] q_3 [M] q_4 [M] q_5 [M] q_6 [M] q_9 [M]	[S] [S] [S] [S] [S] [S] q_7 [S]
i=4	C [S] A [S] q_1 q_2 q_3 q_4 q_5 q_6 [M] q_7 [M] q_9	[S] q_8
i=5	C [S] A [S] q_1 q_2 q_3 q_4 q_5 q_6 q_7 [M] q_8 [M] q_9	[S] [S]

Table 3: The training instance creation example: the importance of a question sentence $[q_1, \dots, q_9]$ is $[q_4, q_6, q_2, q_5, q_3, q_1, q_9, q_7, q_8]$ (from high to low). Six training instances are generated in this example.

	Train	Test	Dev
EQG-RACE	17445	950	1035

Table 4: EQG-RACE Dataset statistics

- **Fluency:** how grammar and structural fluency the generated sentence is.
- **Expectedness:** The extent to which the generated question are in line with expectations.
- **Answerability:** whether the generated question that can be answered.

The human evaluation results are summarized in Table 6. We have the following observations.

For fluency, the two compared models are able to generate grammatical and structural sentences. This is not a surprising result as with the help of the language model, the existing QG models are all able to generate fluent question sentences.

For expectedness, we see there is a big difference between the two compared models. This result validates the KPQG model addresses the QG controllability concern.

For answerability, we also observe improvement. We consider this is due to providing additional keywords guides QG to generate more specific questions other than general questions, which therefore the answerability measure is improved.

3.3 Qualitative Comparison

In Table 7, we show generation results. The examples are selected from the test set of EQG-RACE (Jia et al., 2020). In each example, we show the context paragraph, answer, and the gold question (the first three row of the tables). We use the gold question to simulate it as the one that the user expects to generate. We list the QG results by DeBERTa-QG

and DeBERTa-KPQG with different keyword sets.

Example 1 As can be seen from Example 1, although the result of DeBERTa-QG is the correct question, the direction of the question is not the same as the expected golden question. This is because no keywords are used to guide the QG direction. However, in the results of DeBERTa-KPQG, we can see that with the given [“mars”] keyword, the KPQG model has successfully guided the generation toward the golden question. In addition, KPQG can also use keywords to control the generated sentence syntactical structure. For example, in this case, we prompt [“mars”, “who”] for KPQG. We see that “For conquering plant mars, who did he work with NASA?” is generated. The generated result not only includes the indicated keywords but also consider the order of the keywords. We consider this ability might be also helpful to improve the QG diversity in terms of different syntactical structure generation.

Example 2 In Example 2, we can also see that DeBERTa-KPQG’s question on the given keyword [“largest meat”] is closer to the golden question. Furthermore, prompting different keywords leads to different results. For example, given the [“rice”] keyword, the model generates “Where dos lunch usually eat in order of rice, potatoes and vegetables?”, which is a complete different question direction. This result shows that KPQG can control the generation results according to the keywords given by the user. This feature is also helpful for teachers to have inspiration for preparing reading assessment.

Model	BLEU 1	BLEU 2	BLEU 3	BLEU 4	ROUGE-L	METEOR
(Jia et al., 2020)	35.10	21.08	15.19	11.96	34.24	14.94
BERT-QG	43.37	29.53	22.25	17.54	44.26	20.47
RoBERTa-QG	46.37	32.15	24.34	19.21	46.96	22.32
BART-QG	46.78	32.30	24.53	19.39	47.00	22.22
DeBERTa-QG	47.16	32.81	25.18	20.19	47.33	22.55

Table 5: Performance Comparison

Model	Fluency	Expectedness	Answerability
DeBERTa-QG	1.60	0.86	1.20
DeBERTa-KPQG	1.60	1.37	1.44

Table 6: Human evaluation results

Example 3 Similar to the conclusion from the previous example, in Example 3, we prompt the keyword [“Megan Smith”] to guide the direction of the KPQG model generation. Again, we see the result is close to the golden question. In addition, KPQG can also control the sentence syntax by giving only the “wh-” keyword. For example, in Example 3, the answer is that a person’s name usually uses the sentence structure of “who”, but when the keyword [“which”] is given, KPQG can control the generated result to use “which” as a question syntax. This feature can provide users with the specified sentence syntax when generating questions, helping users to have variability and controllability in the application of generating questions.

4 Conclusion

In this paper, we report the following two findings. First, we find that a very simple QG architecture based on pre-trained language models beats the complicated exam-like QG design (Jia et al., 2020) with or without the keyword indication. Second, by providing keyword information, we can generate results that are closer to the user’s expectation. We believe that our method is more practical to educational QG system applications.

Acknowledgement

This work is supported by MOST 110-2634-F-005-006 - project Smart Sustainable New Agriculture Research Center (SMARTer) and MOST Project under grant No.109-2221-E-005-058-MY3.

References

- Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, et al. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. *arXiv preprint arXiv:2002.12804*.
- Ying-Hong Chan and Yao-Chung Fan. 2019. A recurrent bert-based model for question generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 154–162.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pages 13042–13054.
- Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. *arXiv preprint arXiv:1705.00106*.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Debarta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.
- Xin Jia, Wenjie Zhou, Xu Sun, and Yunfang Wu. 2020. Egg-race: Examination-type question generation. *arXiv preprint arXiv:2012.06106*.
- Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural

Example 1	
Context	At the age of 12, Christopher Hirata already worked on college-level courses, around the time most of us were just in the 7th grade. at the age of 13, this gifted kid became the youngest American to have ever won the gold medal in the International Physics Olympiad. at the age of 16, he was already working with NASA on its project to conquer planet mars. after he was awarded the Ph.D. at Princeton University, he went back to California institute of technology. the next person with very high IQ is Albert Einstein. with an IQ between 160 and 190, Albert Einstein is the genius behind the theory of relativity, which has had great impact on the world of science.
Answer	Christopher Hirata
Gold-Question	Who once worked on the project to conquer planet mars?
DeBERTa-QG	Who was the youngest American to have ever won the gold medal in the International Physics Olympiad?
Keywords 1	“mars”
DeBERTa-KPQG	Who helped NASA on the project to conquer planet mars?
Keywords 2	“mars”, “who”
DeBERTa-KPQG	For conquering planet mars, who did he work with NASA?
Example 2	
Context	Brazil like the French, Brazilians usually eat a light breakfast. Lunch, the largest meal of the day, usually consists of meat, rice, potatoes, beans, and vegetables. between 6:00 p.m. and 8:00 p.m., people enjoy a smaller meal with their families. Brazilians do not mind eating a hurried or light meal and sometimes buy food from street carts. but they always finish eating before walking away.
Answer	Brazil
Gold-Question	In which country do people consider lunch the largest meal?
DeBERTa-QG	Which country has a light breakfast?
Keywords 1	“largest meal”
DeBERTa-KPQG	Which country’s lunch has the largest meal of the day?
Keywords 2	“rice”
DeBERTa-KPQG	Where does lunch usually eat in order of rice, potatoes and vegetables?
Example 3	
Context	Three Central Texas men were honored with the Texas department of public safety’s director’s award in a Tuesday morning ceremony for their heroism in saving the victims of a fiery two car accident. the accident occurred on March 25 when a vehicle lost control while traveling on a rain-soaked state highway 6 near Baylor camp road. it ran into an oncoming vehicle, leaving the occupants trapped inside as both vehicles burst into flames. Bonge was the first on the scene and heard children screaming. he broke through a back window and pulled Mallory Smith, 10, and her sister, Megan Smith, 9, from the wreckage. The girls’ mother, Beckie Smith, was not with them at the time of the wreck, as they were traveling with their baby sitter, Lisa Bow Bin.
Answer	Bonge
Gold-Question	Who saved Megan Smith from the damaged car?
DeBERTa-QG	Who was the first on the scene and heard children screaming?
Keywords 1	“Megan Smith”
DeBERTa-KPQG	Who saved Megan Smith from the accident?
Keywords 2	“which”
DeBERTa-KPQG	In the accident, which man was the hero of the victims?

Table 7: Results of KPQG model

language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2016. Newsqa: A machine comprehension dataset. *arXiv preprint arXiv:1611.09830*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. [Huggingface’s transformers: State-of-the-art natural language processing](#).

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910.

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2017. Neural ques-

tion generation from text: A preliminary study. In *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 662–671. Springer.

Generating Landmark-based Manipulation Instructions from Image Pairs

Sina Zarrieß¹, Henrik Voigt¹, David Schlangen² and Philipp Sadler²

¹University of Bielefeld ²University of Potsdam

¹first.last@uni-bielefeld.de ²first.last@uni-potsdam.de

Abstract

We investigate the problem of generating landmark-based manipulation instructions (e.g. *move the blue block so that it touches the red block on the right*) from image pairs showing a *before* and an *after* state in a visual scene. We present a transformer model with difference attention heads that learns to attend to target and landmark objects in consecutive images via a difference key. Our model outperforms the state-of-the-art for instruction generation on the BLOCKS dataset and particularly improves the accuracy of generated target and landmark references. Furthermore, our model outperforms state-of-the-art models on a difference spotting dataset.

1 Introduction

When speakers produce instructions for tasks in visual environments, they often use landmarks and complex locative expressions to guide listeners to a goal state. Landmarks are well-known to be highly beneficial for achieving communicative success in situated collaborative dialogue tasks like object search, navigation or manipulation (Dräger and Koller, 2012; Clarke et al., 2013). Yet, the accurate generation of landmark-based instructions has been a long-standing challenge in NLG, as it requires complex visual-spatial and linguistic-pragmatic reasoning (Kelleher and Kruijff, 2006). Recent work on *generating* instructions has mostly looked at the navigation domain (Fried et al., 2018; Schumann and Riezler, 2021), whereas work on instruction *following* has shown great interest in manipulation tasks (Bisk et al., 2016; Misra et al., 2017; Shridhar et al., 2020).

In this paper, we investigate the task of generating landmark-based manipulations instructions from image-only input. We use Bisk et al. (2016)’s BLOCKS dataset as it provides both human-generated instructions and corresponding images of a “before state” and an “after state” (see Figure 1).

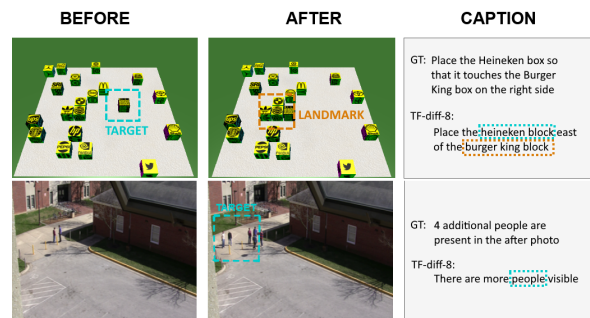


Figure 1: Image-pairs from BLOCKS (top) and Spot-the-diff (bottom) with descriptions generated by our best model. The targets and landmarks are manually highlighted for better view.

We present a transformer-based generation model with a simple but novel difference attention head designed to visually ground complex locative expressions and target-landmark references in image pairs. We show that our model clearly exceeds the performance of Rojowiec et al. (2020)’s existing baseline models on this task, in greatly improving the accuracy of generated target and landmark references. In contrast to other recent instruction generation models (Fried et al., 2017; Köhn et al., 2020; Schumann and Riezler, 2021), our approach does not use any symbolic representations of scene states and trajectories.

A core challenge for instruction generation in our set-up is that the model needs to reason about differences between the “before state” and “after state” represented as an image pair (see Figure 1). As a result of this reasoning, the model should be able to detect the target of the manipulation (e.g. *heineken block*) and verbalizing a suitable description of nearby landmarks (e.g. *east of the burger king block*). We note that the visual reasoning involved here is similar to the problem of spotting image differences or changes, which is a challenging computer vision task (Park et al., 2019; Shi et al., 2020; Oluwasanmi et al., 2019; Gilton et al., 2020).

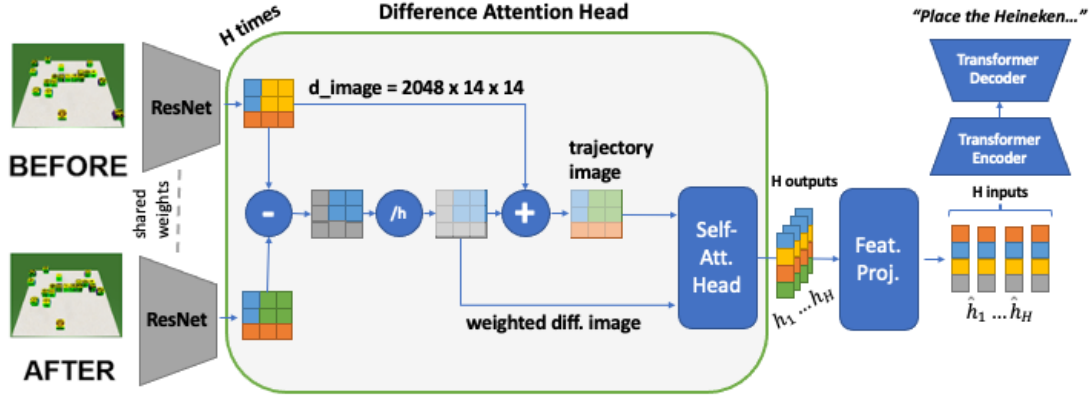


Figure 2: Our difference attention architecture

Thus, for comparison, we use Park et al. (2019)’s model as an additional baseline for instruction generation on BLOCKS. Furthermore, we compare our transformer model against the state-of-the-art on the Spot-the-Diff task with real-word images (Jhamtani and Berg-Kirkpatrick, 2018a).

2 Model

We present a transformer-based model that encodes pairs of *before* and *after* images to generate instructions that describe a particular manipulation to be accomplished in a visual scene. To achieve this, the model needs to learn latent visual-linguistic representations that encode information about the change or manipulation shown in the image pair. As shown in Figure 2, its main idea is a difference attention head that computes an attention map for a visual input state conditioned on the difference to its preceding state in the input.

Our starting point is a vanilla transformer model (Vaswani et al., 2017) that implements self-attention heads, which compute attention maps over values \mathbf{V} given queries \mathbf{Q} and keys \mathbf{K} representing elements of, e.g., a word sequence. A straightforward way to process image pairs with these heads is to allocate two of them: one for the *before* image embedding v_1 and one for the *after* image embedding v_2 .

We propose a difference attention head that exploits an explicit representation of the difference between the two embeddings and set this to \mathbf{K} as a supervision signal that is intended to support the learning of difference-oriented representations. As there is no *before* image for v_1 , we obtain two difference attention heads for an image pair:

- (i) h_1 with $\mathbf{K} = c_1 = 0$ which attends everywhere equally

- (ii) h_2 with $\mathbf{K} = c_2 = v_2 - v_1$ which attends on changes specifically

In line with Park et al. (2019), we scale the output of the difference attention with a trainable parameter γ and apply a residual connection:

$$h_i = \gamma \cdot \text{Attention}(v_i, c_i, v_i) + v_i \quad (1)$$

This simple modification to the keys of the self attention heads takes the idea of difference images from Park et al. (2019) and implements them in a similar way as cross-modal attention in V&L transformers (Tan and Bansal, 2019; Lu et al., 2019).

We hypothesize that, to fully leverage the power of difference attention, more heads, i.e. more visual inputs for a specific change, might be beneficial for grounding and generating utterances. Thus we increase the number of difference attention heads to $H = 8$, where v_H is the *after* image, and we compute “in-between image features” for the additional heads as $v_t = v_1 + c_t$

Intuitively, the “in-between images” represent the trajectory from the *before* to the *after* state (see Figure 2). Formally, we define c_t as the weighted difference features, where the weight is the relative position in the trajectory between v_1 and v_H . Thus, each attention head receives image features representing a different degree of the visual change given by $v_H - v_1$ and accordingly a varying degree of difference features for \mathbf{K} , where the first head at $i = 1$ receives no difference features and the final head at $i = H$ receives the whole difference features as given by the following equation:

$$c_i = \frac{i-1}{H-1} \cdot (v_H - v_1) \text{ where } i \in [1, H] \quad (2)$$

Finally, a single-layer feed forward network maps from the high-dimensional visual image space $2048 \times 14 \times 14$ to the reduced visual word space of 512 dimension $\hat{h}_i = r(h_i)$ and a downstream standard transformer receives the stacked sequence of visual words that represent various levels of change as $V = [\hat{h}_1; \dots; \hat{h}_H]$.

The number of attention heads H is a hyperparameter, which corresponds to the granularity of the simulated visual trajectory $\{v_1, \dots, v_t, \dots, v_H\}$ where later images contain more changes from the *before* image v_1 . We report results for 2 and 8 heads, leaving further experimentation for future work. As baselines, we implement two standard transformers that self-attend to the image pair (**TF-self-att-2**) and to the in-between images (**TF-self-att-8**). These are compared to **TF-diff-att-2** and **TF-diff-att-8** correspondingly, the transformers with difference attention.¹

We encode the *before* and *after* images with a pre-trained ResNet-101 (He et al., 2016) and, optionally, transform it into a sequence with in-between images. This trajectory is passed through a difference attention layer, to obtain a sequence of visual words (see Figure 2). We apply positional encoding to the visual words, as in the standard transformer. These are further processed within the 6 layers of the multi-head-attention-based transformer encoder. In the decoder, an embedding layer first maps the words to vectors and then applies masked-self-attention followed by encoder-decoder attention which relates the visual words to words in the output sequence. In this architecture, difference and self-attention are used consecutively one after the other. In future work, further combinations can be investigated.

3 Experiments

3.1 Data

BLOCKS (Bisk et al., 2016) is a dataset of movement instructions for blocks on a simple virtual 3D board (see Figure 1). The image pairs have been generated by down-sizing MNIST images, decorating the resulting blocks with digits or brand logos and randomly move the block’s pixels to other positions, one at a time. This sequence in reverse order corresponds to an action sequence for assembling a block configuration that visually represents a number. While BLOCKS was originally designed for instruction following, Rojowiec et al.

(2020) analyze its use for instruction giving. We use the MNIST-logo subset with constellations of up to 20 cubes with distinct logos. It is split into 667/95/181 image pairs for training, validation and testing and 6003/855/1629 captions respectively (9 per image pair).

Spot-the-Diff (Jhamtani and Berg-Kirkpatrick, 2018b) provides pairs of similar images extracted from real-word surveillance videos. The image pair shows a scene from the same viewpoint in different, but similar states (according to L_2 distance) resulting in very subtle differences that are difficult to spot. Thus, Jhamtani and Berg-Kirkpatrick (2018b) collected descriptions of these pairs via crowdsourcing and instructed workers to “carefully study the image”, “give sufficient time as some difference may not be obvious” and to provide complete English sentences for each difference. We use the entire dataset of 9524/1634/1404 image-pairs for training, validation and testing and 17676/3310/2107 captions respectively. When an image-pair has less than 3 captions, we re-sample from the given ones, so that during training each pair is seen 3 times per epoch.

3.2 Training and Hyperparameters

We encode the *before* and *after* image separately using a pre-trained ResNet-101 with the last layer cut off which results in image embeddings of size $2048 \times 14 \times 14$ by applying average pooling. The word embedding layer in the transformer decoder is trained from scratch with a size of $d = 512$. We use the Adam optimizer with a learning rate of 10^{-4} and a batch size of 8/16 for training with 8/2 heads respectively. We also perform early stopping after 5 epochs without improvement on the validation set and apply *Label Smoothing* as proposed by Vaswani et al. (2017).

For BLOCKS, it turned out to be necessary to fine-tune the image encoder to recognize the small logos distinguishing the single blocks. The training regime on BLOCKS is a two-stage process: the models (DUDA and our transformer models) are first trained with a frozen, pre-trained image encoder, and then trained fully together to fine-tune the image encoder for this particular task. For Spot-the-diff, we do not fine-tune the image encoder to ensure comparability with previous work.

¹Code <https://github.com/clp-research/diff-att-transformer>

3.3 Evaluation

As the instructions in BLOCKS require detailed descriptions of block configurations, they commonly contain references to target and landmark objects, e.g. *heineken block right of the Burger King block* in Figure 1. If an instruction in BLOCKS does not mention the single correct target, a potential follower will not be able to execute it in any way. For landmarks, there might be several blocks mentioned by different crowd-workers. Since the blocks are generally referred to their logos, the targets in BLOCKS can be detected in human and generated captions with a simple, rule-based instruction parser (Rojowiec et al., 2020). In Spot-the-diff, there might be several target objects referred to by a more complex vocabulary, e.g. *additional people* in Figure 1. The dataset does not provide a language-external annotation for ground-truth target objects and they cannot be easily detected in an automatic way.

We measure the overlap of generated and human captions with BLEU-4, METEOR, CIDEr and SPICE, using the API of Chen et al. (2015). Furthermore, for BLOCKS, we rely on Rojowiec et al. (2020)’s parser which detects expressions (phrases) referring to targets and landmarks in ground-truth and generated instructions. Following Rojowiec et al., we compute these word or phrase accuracies: (i) **target**: correctly generated targets, given all generated target phrases (ii) **landmark**: correctly generated landmarks, mentioning one of the landmarks logos from the set of landmarks found in the ground-truth instructions (iii) **spatial**: correctly generated words not contained in target and landmark phrases, as a simple metric for measuring overlap of spatial expressions.

4 Results

Qualitative samples of generation outputs are shown in Figure 1 and in the Appendix.

4.1 General performance

Table 1 shows the results for instruction generation on BLOCKS: the TF-diff-att-8 transformer achieves the best performance on all metrics. It outperforms the baseline transformers with self attention (TF-self-att-2/8) by a considerable margin. It also clearly improves two state-of-the-art baselines for instruction generation and change captioning. We note that our version of DUDA trained on BLOCKS improves considerably over the results

Model	B	M	C	Target	Landm	Spatial
LSTM+Att*	0.38	0.28	0.27	0.11	0.28	-
DUDA	0.53	0.37	0.96	0.59	0.42	0.66
TF-self-att-2	0.34	0.28	0.35	0.19	0.26	0.76
TF-self-att-8	0.44	0.32	0.66	0.37	0.45	0.72
TF-diff-att-2	0.55	0.38	1.06	0.73	0.40	0.80
TF-diff-att-8	0.68	0.43	1.52	0.86	0.73	0.83

Table 1: BLOCKS results: B(LEU-4), M(eteor), C(ider) and word accuracies (see Section 3.3), LSTM+Att* as reported in Rojowiec et al. (2020).

Model	B	M	C	S
DUDA*	0.081	0.115	0.34	-
FCC*	0.099	0.129	0.368	-
SDCM*	0.098	0.127	0.363	-
DDLA*	0.085	0.12	0.328	-
M-VAM + RAF*	0.111	0.129	0.425	0.171
TF-self-att-2	0.109	0.135	0.777	0.197
TF-self-att-8	0.110	0.136	0.786	0.191
TF-diff-att-2	0.117	0.137	0.843	0.205
TF-diff-att-8	0.113	0.136	0.842	0.202

Table 2: Spot-the-diff results: B(LEU-4), M(eteor), C(IDEr), S(PICE). *Models as reported in Shi et al. (2020)

by Rojowiec et al. (2020), but not over our TF-diff models.

Results on Spot-the-diff are shown in Table 2. Generally, existing systems (mostly developed in the CV community) still obtain relatively low overlap scores on this task (with, e.g., BLEU scores around or below 0.1). Here, again, the difference attention transformers, TF-diff-att-2 and TF-diff-att-8, outperform the vanilla self-attention transformers. They also improve over the state-of-the-art set by the M-VAM model on Spot-the-diff, with a particularly strong increase of the CIDEr score (0.425 and 0.843 respectively). In contrast to BLOCKS, we see a small advantage of the TF-diff-att-2 over TF-diff-att-8. We will discuss this effect in detail in the following Section.

4.2 In-between images and landmarks

Results in Table 1 indicate that the accurate generation of landmark references is a harder task than spotting and referring to target objects. The competitive DUDA model achieves 59% acc. on targets and only 42% acc. on landmarks – an effect which has not been reported in the original DUDA paper by Park et al. (2019). This pattern is expected as the region of the target object is more or less ex-

plicitly represented in the difference image. The landmarks objects, on the other hand, do not move from the *before* to the *after* state and the model has to learn to attend to objects nearby the difference regions.

We observe that in-between images give a very clear performance boost for the realization of landmark references. Thus, the TF-diff-att-8 model improves the landmark accuracy of TF-diff-att-2 and DUDA by more than 30%, cf. Table 1. From this, we conclude that the in-between images combined with difference attention heads allow the transformer model to not only attend to target objects but also to “close-by” landmark objects, i.e. relating the *before* to the *after* image.

On Spot-the-diff, we do not find a clear positive effect of the in-between images, cf. Table 2. However, as discussed in Section 4.1, the differences between models on Spot-the-diff are generally much smaller than on BLOCKS, which likely results from the different nature of the two tasks: the main challenge in Spot-the-diff is to detect and accurately describe extremely small objects, that can be difficult to spot even for humans. At the same time, qualitative inspections of the actual descriptions in Spot-the-diff reveals that they contain much less complex spatial expressions or landmarks. Thus, our results on Spot-the-diff complement rather than contradict results on BLOCKS, and indicate that difference attention with in-between images is particularly helpful for grounding and generating linguistically complex landmark expressions.

4.3 Discussion

Our results are in line with other approaches showing the effectiveness of customized transformer architectures for complex linguistic-visual reasoning (Herdade et al., 2019; Cornia et al., 2020). Our difference attention is tailored to the landmark-based generation task, but generalizes to images from virtual (BLOCKS) and real environments (Spot-the-Diff), and is substantially simpler than, e.g., vision models for difference spotting (Shi et al., 2020). Approaches for video captioning (Zhou et al., 2018; Sun et al., 2019) predict key frames to describe things happening *in* a video with many frames. Our approach is complementary as we augment an image pair with only two frames to obtain in-between frames that are useful for grounding locative expressions and landmarks.

We took inspiration from the DUDA model (Park

et al., 2019) which dynamically attends to *before*, *after* and *difference* images during sequence generation. We carry this idea over to the transformer architecture which attends to all inputs simultaneously, by adding a difference-attention layer that allows the input of fine-granular visual changes between two images at once. Our results show that this approach performs better than dual attention or self-attention alone.

We observe that the different evaluation metrics yield roughly consistent model comparisons, i.e. models with lower overlap scores tend to achieve lower reference-related accuracies. It is worth noting though that the BLEU/Meteor score indicates smaller differences between certain models than the target accuracy: DUDA and TF-diff-att-2 seem to perform almost on par in terms BLEU and Meteor (see Table 1), but the target accuracy indicates that TF-diff-att-2 references are much more accurate. This underlines the fact that n-gram overlap scores in this NLG domain do not constitute a fully satisfactory approximation of instruction quality. An important direction for future work is to design interactive human evaluation settings for these tasks as standard off-line ratings might not be appropriate here (see examples in Appendix for illustration).

5 Conclusion

We investigate language generation for landmark-based instructions, and difference spotting. We proposed a simple difference attention head that relates consecutive images in an input trajectory via a difference key. Our method sets a new state-of-the-art on BLOCKS (Bisk et al., 2016) and Spot-the-diff (Jhamtani and Berg-Kirkpatrick, 2018b). Our findings are in line with Park et al., in that attention mechanisms based on image differences are highly effective for learning to reason for language generation from image pairs. We show that generating instructions with accurate landmark expressions is a challenging task for models at the intersection of Language & Vision, which can be tackled with customized attention mechanisms.

Acknowledgements

We want to thank the anonymous reviewers for their comments. This research/work was partially funded by the *Deutsche Forschungsgemeinschaft* (DFG, German Research Foundation) – 423217434 (RECOLAGE) grant.

References

- Yonatan Bisk, Deniz Yuret, and Daniel Marcu. 2016. [Natural language communication with robots](#). [Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies](#).
- Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server. [arXiv preprint arXiv:1504.00325](#).
- Alasdair Daniel Francis Clarke, Micha Elsner, and Hannah Rohde. 2013. Where’s wally: The influence of visual salience on referring expression generation. [Frontiers in psychology](#), 4:329.
- Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. 2020. Meshed-memory transformer for image captioning. In [Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition](#), pages 10578–10587.
- Markus Dräger and Alexander Koller. 2012. Generation of landmark-based navigation instructions from open-source data. In [Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics](#), pages 757–766.
- Daniel Fried, Jacob Andreas, and Dan Klein. 2017. Unified pragmatic models for generating and following instructions. [arXiv preprint arXiv:1711.04987](#).
- Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. 2018. Speaker-follower models for vision-and-language navigation. In [Advances in Neural Information Processing Systems](#), pages 3314–3325.
- Davis Gilton, R. Luo, R. Willett, and G. Shakhnarovich. 2020. Detection and description of change in visual streams. [ArXiv](#), abs/2003.12633.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In [Proceedings of the IEEE conference on computer vision and pattern recognition](#), pages 770–778.
- Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. 2019. [Image captioning: Transforming objects into words](#). In [Advances in Neural Information Processing Systems](#), volume 32. Curran Associates, Inc.
- Harsh Jhamtani and Taylor Berg-Kirkpatrick. 2018a. Learning to describe differences between pairs of similar images. In [Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing \(EMNLP\)](#).
- Harsh Jhamtani and Taylor Berg-Kirkpatrick. 2018b. Learning to describe differences between pairs of similar images. In [EMNLP](#).
- John Kelleher and Geert-Jan M Kruijff. 2006. Incremental generation of spatial referring expressions in situated dialog. In [Proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the association for computational linguistics](#), pages 1041–1048.
- Arne Köhn, Julia Wichlacz, Álvaro Torralba, Daniel Höller, Jörg Hoffmann, and Alexander Koller. 2020. [Generating instructions at different levels of abstraction](#). In [Proceedings of the 28th International Conference on Computational Linguistics](#), pages 2802–2813, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In [Advances in Neural Information Processing Systems](#), pages 13–23.
- Dipendra Misra, John Langford, and Yoav Artzi. 2017. [Mapping instructions and visual observations to actions with reinforcement learning](#). In [Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing](#), pages 1004–1015, Copenhagen, Denmark. Association for Computational Linguistics.
- Ariyo Oluwasanmi, Muhammad Umar Aftab, Eatedal Alabdulkreem, Bulbula Kumeda, Edward Y. Baagyere, and Zhiquang Qin. 2019. Captionnet: Automatic end-to-end siamese difference captioning model with attention. [IEEE Access](#), 7:106773–106783.
- Dong Huk Park, Trevor Darrell, and Anna Rohrbach. 2019. Robust change captioning. In [Proceedings of the IEEE International Conference on Computer Vision](#), pages 4624–4633.
- Robin Rojowiec, Jana Götze, Philipp Sadler, Henrik Voigt, Sina Zarriß, and David Schlangen. 2020. [From “before” to “after”: Generating natural language instructions from image pairs in a simple visual domain](#). In [Proceedings of the 13th International Conference on Natural Language Generation](#), pages 316–326, Dublin, Ireland. Association for Computational Linguistics.
- Raphael Schumann and Stefan Riezler. 2021. [Generating landmark navigation instructions from maps as a graph-to-text problem](#). In [Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing \(Volume 1: Long Papers\)](#), pages 489–502, Online. Association for Computational Linguistics.

Xiangxi Shi, Xu Yang, Jiuxiang Gu, Shafiq R. Joty, and Jianfei Cai. 2020. Finding it at another side: A viewpoint-adapted matching encoder for change captioning. *ArXiv*, abs/2009.14352.

Kumar Shridhar, Harshil Jain, Akshat Agarwal, and Denis Kleyko. 2020. End to end binarized neural networks for text classification. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 29–34, Online. Association for Computational Linguistics.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. 2019. VideoBERT: A Joint Model for Video and Language Representation Learning. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7463–7472. ISSN: 2380-7504.

Hao Tan and Mohit Bansal. 2019. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111, Hong Kong, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Luowei Zhou, Yingbo Zhou, Jason J. Corso, Richard Socher, and Caiming Xiong. 2018. End-to-End Dense Video Captioning with Masked Transformer. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8739–8748. ISSN: 2575-7075.

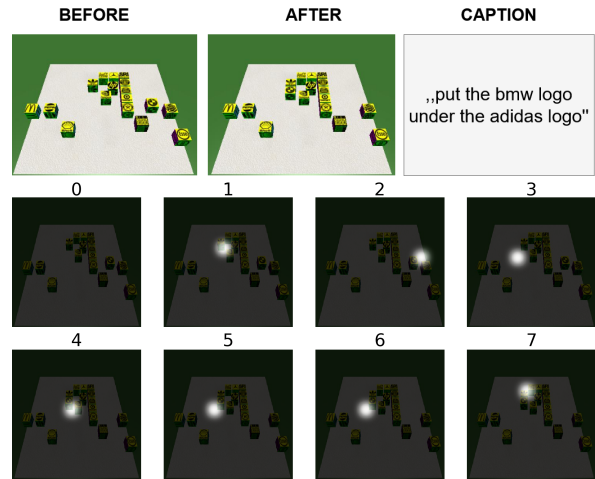


Figure 3: TF-diff-att-8: example caption and attention map on BLOCKS



Figure 4: TF-diff-att-2 attention map on Spot-the-diff for the example from Fig. 1

A Appendix




A.1 Attention maps

Figure 3 shows an attention map for the TF-diff-att-8 model on BLOCKS. The map suggests that the model was able to precisely locate target and landmark objects. Figure 4 shows an example of a very accurate attention map computed by the TF-diff-att-2 model on Spot-the-diff.

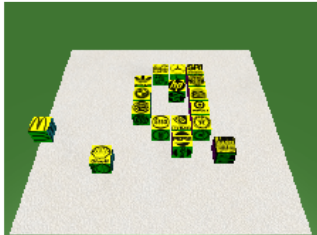

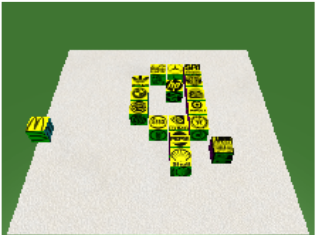
A.2 Qualitative results

We present six generated example outputs drawn from the testset. We show one of the references along with the text generated by the **TF-diff-att-8** model. Additionally, the visual input is presented as the *before* image on the left, the *after* image on the right and an *in-between* image in the middle ($t = 4$).




Example 1

Reference	place the coca cola box so that it's lower right corner touches the upper left corner of the heineken box
TF-diff-att-8	move the coca cola block diagonally above and to the left of the heineken block.
	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>8_num0_16.png</p>  </div> <div style="text-align: center;"> <p>8_num0_17.png</p>  </div> <div style="text-align: center;"> <p>8_num0_17.png</p>  </div> </div>




Example 2

Reference	shell moves two spaces left and 3/4 space down from heineken.
TF-diff-att-8	place the shell block south of the pepsi block.
	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>99_num9_13.png</p>  </div> <div style="text-align: center;"> <p>99_num9_14.png</p>  </div> <div style="text-align: center;"> <p>99_num9_14.png</p>  </div> </div>




Example 3 (wrong landmark)

Reference	place adidas so it's right side completely touches coca-cola.
TF-diff-att-8	put the adidas block in the first open space to the left of the mercedes block.
	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>39_num3_19.png</p>  </div> <div style="text-align: center;"> <p>39_num3_20.png</p>  </div> <div style="text-align: center;"> <p>39_num3_20.png</p>  </div> </div>




Example 4

Reference	esso is placed so that its upper right corner is touching the lower left corner of nvidia
TF-diff-att-8	move esso so it is below and to the left of nvidia
	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>88_num8_08.png</p>  </div> <div style="text-align: center;"> <p>88_num8_09.png</p>  </div> <div style="text-align: center;"> <p>88_num8_09.png</p>  </div> </div>

Example 5 (wrong)

Reference	place the esso block in the same column as the coca cola block, one and a half block spaces above the coca cola block.
TF-diff-att-8	place the shell block north of the stella artois block.
	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>8_num0_00.png</p>  </div> <div style="text-align: center;"> <p>8_num0_01.png</p>  </div> <div style="text-align: center;"> <p>8_num0_01.png</p>  </div> </div>

Example 6 (wrong landmark)

Reference	stella artois is placed directly to the left of texaco
TF-diff-att-8	move the stella artois block to the left of the target block.
	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>88_num8_06.png</p>  </div> <div style="text-align: center;"> <p>88_num8_07.png</p>  </div> <div style="text-align: center;"> <p>88_num8_07.png</p>  </div> </div>

Zero-shot Cross-Linguistic Learning of Event Semantics

Malihe Alikhani¹ Thomas Kober² Bashar Alhafni^{*3} Yue Chen^{*4} Mert Inan^{*1}
Elizabeth Nielsen^{*2} Shahab Raji^{*5} Mark Steedman² Matthew Stone⁵

¹University of Pittsburgh ²University of Edinburgh ³New York University Abu Dhabi
⁴Indiana University ⁵Rutgers University

Abstract

Typologically diverse languages offer systems of lexical and grammatical aspect that allow speakers to focus on facets of event structure in ways that comport with the specific communicative setting and discourse constraints they face. In this paper, we look specifically at captions of images across Arabic, Chinese, Farsi, German, Russian, and Turkish and describe a computational model for predicting lexical aspects. Despite the heterogeneity of these languages, and the salient invocation of distinctive linguistic resources across their caption corpora, speakers of these languages show surprising similarities in the ways they frame image content. We leverage this observation for zero-shot cross-lingual learning and show that lexical aspects can be predicted for a given language despite not having observed any annotated data for this language at all.

1 Introduction

Tense and aspect rank among the most ubiquitous, problematic, and theoretically vexed features of natural language meaning (Hamm and Bott, 2018). Systems of tense and aspect differ considerably—but also often quite subtly—across languages. Figure 1 shows how the corpus manifests differences and similarities across languages that align with their grammatical structures. Tense and aspect have received extensive study across cognitive science; see Hamm and Bott (2018). Nevertheless, from a computational point of view, it has been extremely challenging to gain empirical traction on key questions about them: how can we build models that ground speakers’ choices of tense and aspect in real-world information? how can we build models that link speakers’ choices of tense and aspect to their communicative goals and the discourse context? how can we build models that recognize

tense and aspect? This is particularly challenging because we might have to work with small annotated datasets. The data scarcity issue renders the need for effective cross-lingual transfer strategies: how can one exploit abundant labeled data from resource-rich languages to make predictions in low resource languages?

In this work, we leverage image descriptions to offer new insights into these questions. For the first time, we present a dataset of image descriptions and Wikipedia sentences annotated with lexical aspects in six languages. We hypothesize that across all of the languages that we study, image descriptions show strong preferences for specific tense, aspect, lexical aspect, and semantic field. We adapt the crowdsourcing methodology used to collect English caption corpora such as MSCOCO and Flickr (Young et al., 2014; Lin et al., 2014) to create comparable corpora of Arabic, Chinese, Farsi, German, Russian, and Turkish image captions. We extend the methodology of Alikhani and Stone (2019) to get a synoptic view of tense, lexical aspect, and grammatical aspect in image descriptions in these diverse languages.

Finally, we study the extent to which verb aspect can be predicted from distributional semantic representations across different languages when the model was never exposed to any data of the target language during training, essentially performing zero-shot cross-lingual transfer. We consider predicting lexical aspect at the phrase level an important prerequisite for modelling fine grained entailment relations, such as inferring consequent states (Moens and Steedman, 1988). For example, this is important for keeping knowledge bases up-to-date by inferring that the consequence of *Microsoft having acquired GitHub*, is that now, *Microsoft owns GitHub*.

Our results show that the grammatical structure of each language impacts how caption information is presented. Throughout our data, we find, as in

* Equal contribution.


	<p>Arabic</p> <p>رجل يمشي بجانب الطريق. street nearby walking-PRS-MASC-IPFV-3SG man A man is walking nearby the street.</p>
	<p>Chinese</p> <p>雙層公共汽車正在公路上行駛 double-decker public bus now IPFV road on drive Double-decker public buses are driving on the road.</p>
	<p>Farsi</p> <p>اتوبوس‌های دوطبقه در خیابان حرکت می‌کنند. do move street in double-decker bus-PL Double-decker buses are moving in the street.</p>
	<p>German</p> <p>Zwei Busse fahren an einer Haltestelle vorbei. Two buses drive a bus stop past. Two buses drive past a bus stop.</p>

Figure 1: An example image from the MSCOCO dataset with Arabic, Chinese, German and Farsi captions. (ID: 000000568439, photo credit: Stephen Day)

Figure 1, that captions report directly visible events, focusing on what’s currently in progress rather than how those events must have begun or will culminate. Yet they do so with different grammatical categories across languages: the progressive aspect of Arabic; the unmarked present of German; or the aspectual marker of the imperfective verbs of Chinese describing an event as in progress.

2 Related Work

Linguists and computational linguists have largely focused on aspectuality as it has been used in unimodal communication. Caselli and Quochi (2007) showed how aspectual information plays a crucial role in computational semantic and discourse analyses. Pustejovsky et al. (2010) described how aspect must be considered for event annotations and Baiamonte et al. (2016) incorporated lexical aspect in the study of the rhetorical structure of text. Kober et al. (2020) presented a supervised model for studying aspectuality in unimodal scenarios only in English. In this work however, we focus on image captions that enable us to better understand how humans describe images. We also explore for the first time the potential of zero-shot models for learning lexical aspect across languages and genre.

The field of automatic image description saw an explosive growth with the release of the Flickr30K and MSCOCO datasets (Vinyals et al., 2015). Fewer works however, have studied how humans produce image descriptions (Bernardi et al., 2016; Li et al., 2019). For example, van Miltenburg et al. (2018a) studied the correlations between eye-gaze patterns and image descriptions in Dutch. Jas and Parikh (2015) investigated the possibility of predict-

ing image specificity from eye-tracking data and van Miltenburg et al. (2018b) discussed linguistics differences between written and spoken image descriptions. In this work we continue this effort by offering the first comparative study of verb use in image description corpora that we have put together in six different languages. Alikhani et al. (2020); McCloud (1993); Cohn (2013); Alikhani and Stone (2018); Cumming et al. (2017); Alikhani et al. (2019) proposed that the intended contributions and inferences in multimodal discourse can be characterized as coherence relations. Our analyses and computational experiments explore the extent to which different grammatical-based distinctions correlate with discourse goals and contextual constraints and how these findings generalize across languages.

3 Data Collection and Annotation

Given a set of images, subjects were requested to describe the images using the guideline that was used for collecting data for MSCOCO (Lin et al., 2014). The instructions were translated to six target languages. For the Chinese instructions, we reduced the character limits from 100 to 20 since the average letter per word for English is 4.5. Generally, a concept that can be described in one word in English can also be described in one or two characters in Chinese. The original guideline in English as well as the translations can be found in the attached supplementary material.

We recruited participants through Amazon Mechanical Turk and Upwork.¹ All subjects agreed to a consent form and were compensated at an esti-

¹<https://www.upwork.com/>

mated rate of USD 20 an hour. We collected captions for 500 unique images (one caption per image in each of the languages that we study in this paper) that were randomly sampled from MSCOCO for each language. The results of our power analysis suggest that with this sample size, we are able to detect effects sizes as small as 0.1650 in different distributions of lexical aspect with a significance level of 95% (Faul et al., 2014).

Annotation Effort. The data is annotated by expert annotators for language specific characteristics of verbs such as tense, grammatical and lexical aspect and the Cohen Kappa inter-rater agreements (Cantor, 1996) are substantial ($\kappa > 0.8$) inter-annotator agreement across the languages.

3.1 Methods

To compare captions and text in a different unimodal genre, we randomly selected 200 sentences across all languages from Wikipedia and annotated their lexical aspect. For Arabic, we used MADAMIRA (Pasha et al., 2014) to analyze the image captions which are written in Modern Standard Arabic. We limited the 200 Chinese Wikipedia sentences to 20 characters in length. The word segmentation and part-of-speech tagging are performed using Jieba Python Chinese word segmentation module (Sun, 2012). Traditional Chinese to Simplified Chinese character set conversion was done using zhconv.²

The Farsi image captions and the Wikipedia sentences were automatically parsed using *Hazm* library. For German, we used UDPipe (Straka and Straková, 2017) and we have analysed the Russian morphological patterns by pymorphy2 (Korobov, 2015). For Turkish, the morphological analysis of all the verb phrases in the Wikipedia sentences and the captions are performed using the detailed analysis in (Ofazer et al., 1994). While separating noun phrases from verb phrases, stative noun-verbs of existence (“var” instead of “var olmak”) were considered as verbs as well, following the analysis by (Çakmak, 2013).

4 Data Analysis

We performed an analysis of our data to study the following questions: What do image descriptions in Arabic, Chinese, Farsi, German, Russian and Turkish have in common? What are some of the

language-specific properties? What opportunities do these languages provide for describing the content of images? In what follows, we first describe similarities across languages. Next we discuss language specific properties related to tense and aspect.

In general, captions are less diverse as opposed to Wikipedia verb phrases in terms of their verbs vocabulary across the six languages. Table 1 shows the accumulative percentage of top K verbs for the six languages for Wikipedia and image captions. Wikipedia sentences and captions have different distributions of tense, grammatical aspect and lexical aspect across all languages ($p < 0.01$, $\chi > 12.5$). When it comes to Arabic, atelic verbs dominate the verbs used in Arabic captions. However, the stative verbs dominate the verbs used in Wikipedia sentences.

Moreover, present imperfective verbs make 99% and present perfective verbs make 1% of 85 inflected verbs across all Arabic captions. However, this is drastically different in our baseline. Across 200 full Arabic Wikipedia sentences and out of 180 inflected verbs, present perfective and present imperfective make 49.5% and 2% respectively. Whereas, past perfective and past imperfective make 44.6% and 4% respectively.

This largely agrees with what we analyzed for other languages. In the Chinese data, 56% of Chinese caption verbs are imperfective whereas the majority (70%) of the Chinese Wikipedia descriptions are stative. Chinese Wikipedia sentences also have very few atelic descriptions (1.8%) whereas Chinese captions are populated with atelic descriptions. Chinese does not have tense, but we annotated the sentences both in captions and Wikipedia to learn about the number of sentences that present some kind of cues to refer to an event in the past i.e. adverb. In Wikipedia, 26% of sentences refer to events in past but this number decreases to less than 1% in captions. For Farsi, atelic events make up to 72% of Farsi captions and 17% of Farsi Wikipedia. As in Arabic and Chinese, we observed a major difference in distributions of grammatical aspect and tense in Farsi Wikipedia and Farsi captions. Farsi captions are populated with simple and imperfective present verbs. German captions also follow the general trend with 96% of verbs in caption exhibiting imperfective aspect, in comparison to only 57% in Wikipedia. Atelic verbs dominate the Aktionsart distribution of the captions dataset, making up 55%

²<https://github.com/gumbllex/zhconv>

	Arabic		Chinese		Farsi		German		Russian		Turkish	
	Wiki	Capt.	Wiki	Capt.	Wiki	Capt.	Wiki	Capt.	Wiki	Capt.	Wiki	Capt.
Top 10	0.262	0.688	0.264	0.367	0.364	0.664	0.394	0.582	0.257	0.654	0.283	0.457
Top 30	0.485	0.937	0.396	0.589	0.466	0.854	0.567	0.804	0.455	0.900	0.524	0.666
Top 100	0.832	–	0.650	0.911	0.545	–	0.911	–	0.802	–	0.728	0.856

Table 1: Captions show a limited distribution of verbs in comparison with Wikipedia. Verb use in Chinese and Turkish captions dataset are more diverse than in Farsi and Arabic caption datasets.

of all verb occurrences, whereas only 16% of verbs are atelic in the Wikipedia sample. The trend is conversed for telic verb occurrences, which make up only 4% in the captions dataset, but 43% in the Wikipedia sample. Interestingly, the proportion of stative verbs is roughly equal in captions and Wikipedia.

The Russian data also hold with these general trends: all captions are imperfective, whereas only 50% of Wikipedia sentences are. This distribution is even more extreme in Russian than in other languages partially because of a unique property of the Russian aspectual and tense system: only verbs that refer to past or future events in Russian can be perfective. In the captions, 99% of verbs refer to present events and therefore are required to be imperfective. This also is borne out the telicity of Russian captions: 49% of captions are atelic, 30% are stative, and only 22% are telic. By contrast, only 21% of Wikipedia data is atelic, while 26% is stative, and 53% is telic. As discussed in Section 4.1 below, this reflects a correlation between perfectivity and telicity in Russian.

Telicity of the Turkish data follows a similar distribution to the other languages, with a key difference in the statistics of stative verbs. Both Wikipedia sentences and captions have higher count of stative verbs compared to other languages. 56% of Wikipedia verbs and 63% of caption verbs are stative in Turkish. This is caused by the inherent copula usage and preference of stative and timeless tenses such as the “geniş zaman”. Atelic verb percentage in captions (30.4%) is considerably smaller to that of stative verbs (63.8%). There is a drastic difference between the number of telic verbs with a 32.4% in Wikipedia phrases compared to 5.8% in captions.

4.1 Language-Specific Observations

Arabic. Arabic has a rich morphological system (Habash, 2010). Moreover, verbs in Arabic have three grammatical aspects: perfective, imperfective, and imperative. The perfective aspect indicates that

actions described are completed as opposed to the imperfective aspect which does not specify any such information. Whereas the imperative aspect is the command form of the verb.

Similar to German and Russian, non-past imperfective verbs were dominant across the captions in Arabic as opposed to Chinese, Farsi, and Turkish. Furthermore and as shown in Table 2, 72.2% of Arabic captions were atelic, and this is the highest atelic percentage for captions across all languages. Whereas, 8.9% of the Arabic Wikipedia sentences were atelic, which constitutes the lowest atelic percentage for Wikipedia sentences across all other languages. This highlights an interesting evidence of the morphological richness in Arabic and how verbs can inflect for mood and aspect.

Chinese. Chinese is an equipollent-framed language (E-framed language), due to its prominent feature – serial verb construction (Slobin, 2004). For example, 走进 (walk into) and 走出 (walk out of) are treated as two different verbs. This phenomenon greatly enlarged the vocabulary of Chinese verbs perceived by POS taggers and parsers. We believe this is an important reason why Chinese verbs look so diverse and the distribution among atelic, telic and stative looks rather imbalanced. Having the base verb character and adding on aspectual particles changes the telicity. Given the nature of Wikipedia text, it is observed that in table 2 only 1.8% are atelic and more than 69.8% are stative, while in image captions more than 56% are atelic.

Since Chinese does not have the grammatical category of tense, the concept denoted by tense in other languages is indicated by content words like adverbs of time or it is simply implied by context. For example, the verb for “do” is 做 (zuo), which is used to describe all past, present, and future events. Since the verb remains the same, temporal reference is instead indicated by the time expressions (Lin, 2006), for example:

- (1) 昨天 我做了 批萨。

	Arabic		Chinese		Farsi		German		Russian		Turkish	
	Wiki	Capt.	Wiki	Capt.	Wiki	Capt.	Wiki	Capt.	Wiki	Capt.	Wiki	Capt.
Atelic	0.089	0.722	0.018	0.561	0.171	0.719	0.162	0.550	0.213	0.488	0.114	0.304
Telic	0.371	0.010	0.285	0.063	0.470	0.042	0.431	0.038	0.530	0.218	0.324	0.058
Stative	0.540	0.268	0.698	0.377	0.357	0.237	0.407	0.412	0.257	0.299	0.560	0.638

Table 2: Captions include more atelic descriptions in comparison with Wikipedia across languages.

Yesterday I do PFV pizza.
 Yesterday I made pizza.

Farsi. In the Farsi caption dataset four verbs make up to around 50% of the verbs: *to be* (بودن), *to play* (بازی کردن), *to sit* (نشستن), and *to look* (نگاه کردن)

Table 1 shows difference in verbs distributions across languages. The data regarding the distribution of caption verbs in English are reported by (Alikhani and Stone, 2019). Chinese captions are much more diverse and the difference is statistically significant ($p < 0.05$, $\chi = 14.4$).

Farsi verbs are either simple or compound. Any lexical unit which contains only a verbal root is a simple verb (e.g. verbal root: رفتن ‘to go’). The lexical unit which contain either a prefix plus a verbal root, or a nominal plus either a regular verbal root or an auxiliary verb are compound verbs. Related to this is the phenomenon of incorporation, defined by (Spencer, 1991) as the situation in which “a word forms a kind of compound with its direct object, or adverbial modifiers while retaining its original syntactic function.”

59.3% of Farsi Caption verbs are compound and 88.2% of the compound verbs are constructed with کردن (to do) and شدن (to be). Wikipedia on the other hand includes only 12.1% compound verbs. Majidi (2011) conjectured that کردن (to do) and شدن (to be) are used when the speaker wants to highlight the meaning of the noun even more in comparison with cases where nouns are accompanied with گرفتن (to take) or داشتن (to have). For example, نگاه کردن (literally *Do a look*) is the fourth most frequent verb in captions.

However, the majority (97%) of the compound verbs in captions are constructed with nouns.

Megerdoomian (2002) hypothesized that the aspectual properties depend on the interaction between the non-verbal and the light verb and that the choice of light verb affects argument structure. For instance, to form the transitive version of an intransitive predicate, Farsi speakers replace the light verb by its causative form. **All of the intransitive**

compound verbs in our corpus are atelic.

German. German speakers predominantly used the present simple — rather than the present progressive — to describe atelic activities, where we found that only $\approx 7\%$ of atelic captions have been described in the present progressive. For example, sentences (1)-(2) below show two captions where the ongoing activity is described in the present simple in German, however in English, the present progressive would be used. In English, the use of the present simple has a strong futurate reading, which is substantially weaker in German. Thus we attribute the frequent use of the present simple in German to it being less aspectually ambiguous.

- (1) Zwei Männer **spielen** Wii im Wohnzimmer.
*Two men **are playing** on a Wii in the living room.*
- (2) Ein Mann und eine Frau **fahren** Ski.
*A man and a woman **are skiing**.*

We furthermore found that German speakers have frequently omitted the verb altogether if an imaged depicted some form of still life. These sentences exhibit stative lexical aspect, and typically, verbs such as “stand”, “lie” or a form of “to be” would have been the correct verb as sentences (3)-(4) below demonstrate, where we have added a plausible verb in square brackets.

- (3) Ein Zug [**steht**] neben einer Ladeplattform.
*A train [**is standing**] next to a loading bay.*
- (4) Eine Pepperoni Pizza [**liegt**] in einer Pfanne neben einem Bier.
*A pepperoni pizza [**is lying**] in a pan next to a beer.*

Russian. A distinction between imperfective and perfective aspect must be marked on all Russian verbs. This contrasts with languages (e.g., Spanish) where aspect is only marked explicitly in a subset of the verbal system, such as within the past tense.

Aspect marking in Russian is often done by means of affixation: a default-imperfective stem becomes perfective with the addition of a prefix (e.g. *pisat'* > *napisat'* 'to write' (Laleko, 2008)). Perfective aspect expresses a view of an event "in its entirety" (Comrie, 1976), including its end point, meaning that perfectivity and telicity are highly correlated. For example, the use of the perfective *napisat'* 'to write' implies the completion of a finite amount of writing, whether or not the speaker chooses to include an explicit direct object indicating what is being written. There is disagreement in the literature on whether all perfective verbs in Russian are telic or if the perfectivity is merely correlated with telicity (Guéron, 2008; Filip, 2004). However, the fact that all verbs must be explicitly marked as either perfective or imperfective, combined with the fact that telicity is at least positively correlated with perfectivity, may lead to more verbs in the Russian being labelled as telic. In fact, we do find that when compared with languages such as English, where verbs may remain under-specified for aspect and therefore for telicity, the Russian captions contain significantly more telic verbs.

Turkish. Lexical aspects of verbs in Turkish captions differ from other languages in terms of choice of the sentence structure and the diversity of Turkish tenses, with the presence of copula. These intricacies are analyzed using the work of (Aksan and Aksan, 2006) and (Aksan, 2003). It can be observed that Turkish-speakers tend to choose a specific sentence structure while describing pictures.

Captions are populated with noun phrases consisting of a verbal adjective, a subject and an implicit noun-verb ("var"). The most important aspect about determining lexical aspect in Turkish is the plethora of tenses. A considerably different tense is the "geniş zaman", which translates to "broad time/tense". Its use broadens the time aspect in a verb to an extent that the verb exists in a timeless space. Even though it is generally compared with the present simple tense in English, "geniş zaman" telicity greatly depends on the context and the preceding tense in the agglutinative verb structure. Wikipedia sentences contain 13.3% "geniş zaman" verbs while caption verbs do not have any of that formation. This is due to the difference of giving a description or a definition.

Turkish definitions are timeless and use "geniş zaman" more frequently, while descriptions, like

in the captions, use other tenses. It can be presumed that all "geniş zaman" verbs are atelic; however, this does not necessarily hold true in captions where a limited number of telic cases exist, which increases the importance of a differentiation between atelic and telic tenses in Turkish. Another distinction that is visible between the Turkish image captions and Wikipedia sentences is the progressive aspect. 59.7% of caption verbs are progressive while only 0.9% of Wikipedia verbs are progressive. This aspect is used extensively in captions due to its close relation with any action verb that is being done.

5 Computational Experiments

In this section we leverage our multilingual annotated dataset and investigate to what extent aspect can be detected with computational methods. More specifically, the primary research question we address in this section is an empirical investigation whether distributional semantic models capture enough information about the latent semantics of aspect to be detected across languages.

Our use of distributional semantic representations is furthermore motivated by the fact that they are readily available in numerous languages, and that they, contrary to manually constructed lexicons such as VerbNet (Schuler and Palmer, 2005) or LCS (Dorr and Olsen, 1997), scale well with growing amounts of data and across different languages. Furthermore, there is a growing body of evidence that models based on the distributional hypothesis capture some facets of aspect (Kober et al., 2020; Metheniti et al., 2022), despite the fact that aspect is represented in a very diverse manner across languages.

5.1 Aspectual Classification

We treat the prediction of verb aspect as a supervised classification task and experiment with pre-trained fastText (Grave et al., 2018) embeddings³, multilingual BERT (Devlin et al., 2019), and ELMo (Peters et al., 2018; Che et al., 2018)⁴ as input, and the aspectual classes *state*, *telic*, *atelic* as targets. For fastText we average the word embeddings to create a single vector representation, for

³<https://fasttext.cc/docs/en/pretrained-vectors.html>

⁴While the BERT model is truly multilingual, we use a single monolingual ELMo model for our experiments from <https://github.com/HIT-SCIR/ELMoForManyLangs>.

Aspect	Arabic		Chinese		Farsi		German		Russian		Turkish		
	Capt.	Wiki	Capt.	Wiki	Capt.	Wiki	Capt.	Wiki	Capt.	Wiki	Capt.	Wiki	
fastText	Atelic	0.95	-	0.97	-	0.95	-	0.90	-	0.96	-	0.51	-
	Telic	-	0.48	-	0.00	-	0.74	-	0.89	-	0.83	-	0.62
	State	0.84	0.66	0.00	0.89	0.83	0.59	0.88	0.88	0.94	0.27	0.83	0.80
mBERT	Atelic	0.50	-	0.80	-	0.73	-	0.72	-	0.78	-	0.96	-
	Telic	-	0.64	-	0.92	-	0.75	-	0.84	-	0.83	-	0.79
	State	0.88	0.79	0.91	0.47	0.93	0.57	0.82	0.82	0.88	0.44	0.91	0.89
ELMo	Atelic	0.65	-	0.76	-	0.77	-	0.78	-	0.90	-	0.97	-
	Telic	-	0.66	-	0.87	-	0.79	-	0.76	-	0.83	-	0.74
	State	0.89	0.78	0.88	0.22	0.93	0.67	0.85	0.75	0.94	0.20	0.93	0.86

Table 3: Mono-lingual F1-scores per label across all languages with using fastText embeddings (top), multilingual BERT embeddings (middle) and ELMo embeddings (bottom).

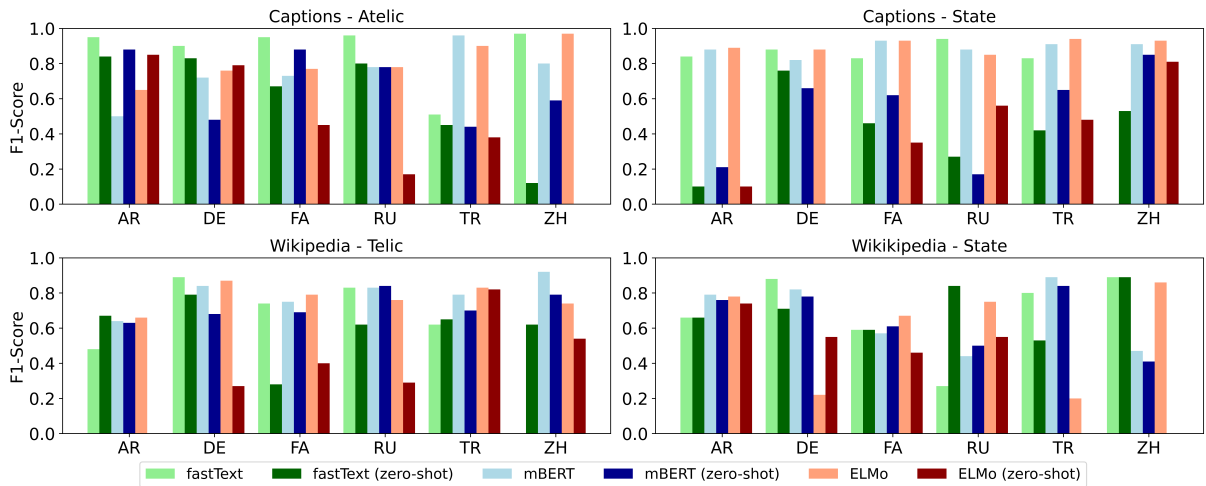


Figure 2: Performance comparison between zero-shot cross-lingual (darker shades) learning and a mono-lingual (lighter shades) setup. Remarkably, even without any target language data, our simple zero-shot setup is competitive with using mono-lingual data and even surpasses it in some cases.

multilingual BERT we use its [CLS] token, and for ELMo the pooled representation of the encoded utterance for classification. We use the Logistic Regression classifier from scikit-learn (Pedregosa et al., 2011) with default hyperparameter settings.

Our choice of models is motivated by: a) assessing performance with a word-level model (fastText), b) estimating the performance difference when large pre-trained models (ELMo & mBERT) are applied, and c) observing the difference between a single multilingual model (mBERT) and monolingual models for the different languages (fastText & ELMo).

Mono-lingual. For the mono-lingual experiments, we evaluate our method on the annotated captions and Wikipedia sentences, however we decided to drop all *telic* instances from the captions

data, and all *atelic* instances the Wikipedia sentences, as they occur very infrequently in either respective corpus.⁵ We are focused on establishing whether aspect can be predicted from embeddings across languages *in principle* and wanted to avoid obfuscating the problem of predicting aspect with the problem of class imbalance.

The aim of our first experiment is to establish that aspect can be classified for our set of languages with distributional representations in a supervised setting as has been shown on English data (Kober et al., 2020). Figure 3 shows the difference in Accuracy of our models in comparison to a majority class baseline. As the figure shows, the distributional models are able to outperform the majority

⁵This reduced the classification problem to a 2-class problem, *stative vs. non-stative*.

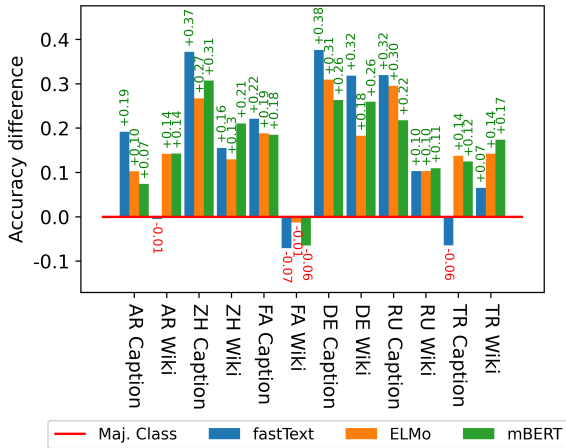


Figure 3: Accuracy comparison of a majority class baseline to fastText, multilingual BERT and ELMo models across all our target languages and domains.

class baseline by substantial margins across the board with the exception of our Farsi Wikipedia dataset where we underperform the baseline by a small margin.

Next, we aim to establish baseline scores for the distributional models on our dataset. We perform stratified 10-fold cross-validation on our annotated datasets and report a micro-averaged F1-Score on the basis of accumulating the number of true positives, false positive, and false negatives across all cross-validation runs (Forman and Scholz, 2010).

Table 3 shows that except for Chinese, our simple method of predicting aspect from averaged fastText embeddings works astonishingly well across languages, achieving F1-scores in the mid-80s to mid-90s for many languages. Multilingual BERT and ELMo perform similarly across languages with notable problems for distinguishing between states and *telic* events in Russian and Chinese.

Overall, all models perform approximately in the same ballpark, specifically, there is no dramatic loss in performance when using a single multilingual model in comparison to monolingual models. Conversely, an LSTM-based model and the even simpler bag-of-words based model work remarkably well given the latent nature of aspect. Distributional representations appear to capture enough information for making fine-grained semantic distinctions — an important result for further work on multilingual semantic inference around consequence and causation (Mirza and Tonelli, 2014; Kober et al., 2019; Guillou et al., 2020).

Zero-Shot Cross-lingual. For the zero-shot cross-lingual experiment we use the aligned fast-

Text embeddings and the same mBERT and ELMo models as in the mono-lingual experiments.⁶ We perform a zero-shot learning on the basis of a leave-one-language-out evaluation. This means that we train our Logistic Regression classifier on the data of five languages and evaluate performance on the sixth one. The models were never exposed to *any* data of the target language during training, thereby performing zero-shot cross-lingual transfer. This assesses how much information can be leveraged cross-lingually, which has potential further applications for transfer learning and data augmentation.

As for the mono-lingual experiments we drop the *telic* class from the captions data, and the *atelic* class from the Wikipedia data. Figure 2 compares mono-lingual with zero-shot cross-lingual performance, showing that our simple setup yields remarkably strong results, that in some cases even outperform the mono-lingual setup. Our results indicate that a considerable amount of aspectual information can be transferred and induced cross-lingually, providing a very promising avenue for future work.⁷ In order to estimate the importance of the contribution of each language in the zero-shot setting we conduct a Shapely-flavoured (Shapley, 1953) analysis. Shapely values are a method for quantifying the contribution to model performance of any given feature in a dataset (Molnar, 2022).

As Shapely values operate on the *feature* space, rather than the *instance* space, we interpret the presence of training data for a particular languages as a binary indicator feature. This means that any languages can be “active” during model training, or not. This way, we can observe the performance of a model with and without any given language in the training data, and estimate that language’s impact on model performance. The process to estimate the Shapely-flavoured impact value for a given language is perhaps best explained by an example: supposing our target language — for which we want to predict aspect — is Arabic, and we want to quantify the contribution of German language training data in our model, we start by training a model on Farsi data and compare our model’s predictive performance to a model trained on Farsi *and* German data. Next, we train our model on Farsi and Russian data, and compare its performance to a model trained on Farsi, Russian *and*

⁶<https://fasttext.cc/docs/en/aligned-vectors.html>

⁷A multilingual companion table to Table 3 is presented in Table 5 in Appendix 7.

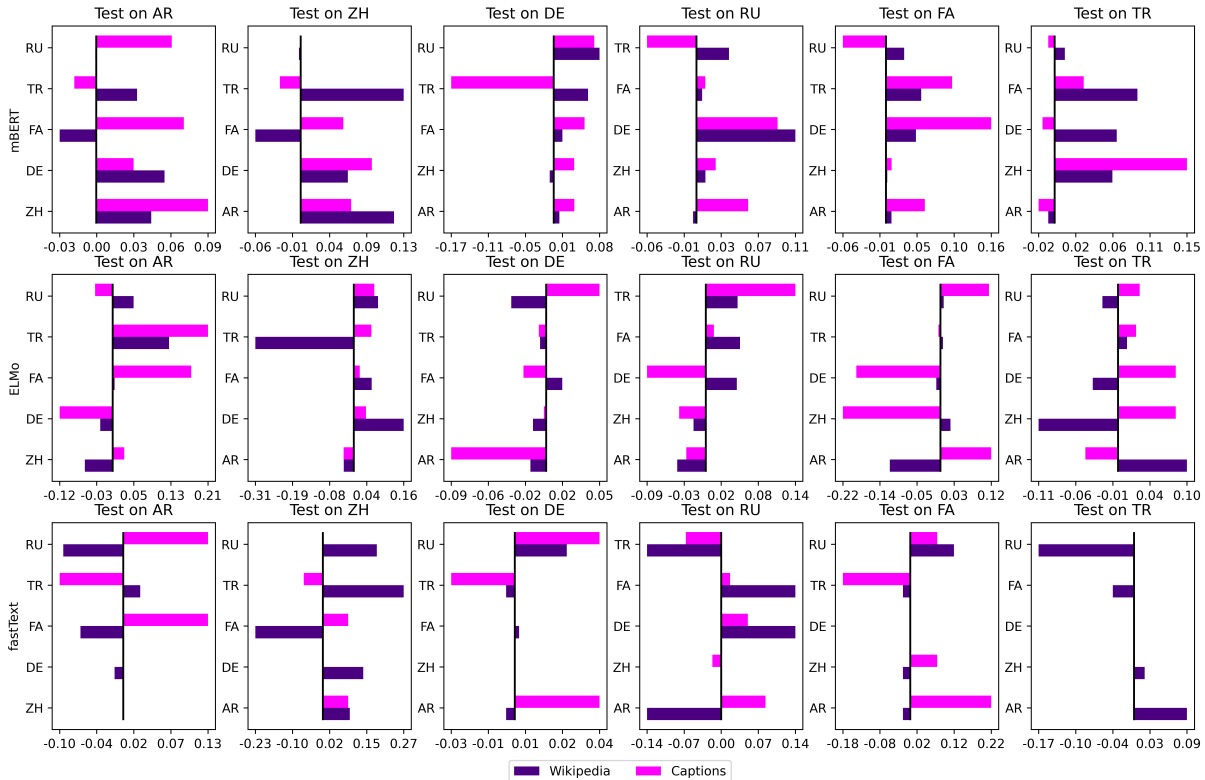


Figure 4: Shapely-flavoured analysis of the impact of each language’s presence in the training data on predicting aspect in a target language in a zero-shot cross-lingual setting.

German data, and so on for all combinations of training data. Lastly, we average the differences of all these comparisons to obtain a value that represents the impact of German data on predicting aspect for Arabic. We perform this method for all model, language and domain combinations, with the resulting Figure 4 summarising all Shapely-flavoured impact values for all languages. The figure shows the positive and negative impact of each language — for the captions dataset in magenta and the Wikipedia dataset in indigo — for measuring accuracy. Generally, the impact of each language on model performance is primarily governed by the *kind of model*, rather than the language(s) used for training. While this may seem somewhat dissatisfying at first, we believe that understanding model behaviour is paramount for transfer learning with cross-lingual data with the goal of leveraging e.g. the explicit aspectual markers in the Slavic languages to learn models for languages such as English where aspect is more opaque, as a very fruitful avenue for future research.

6 Conclusion

By analyzing verb usage in image–caption corpora in Arabic, Chinese, Farsi, German, Russian and

Turkish we find that people describe visible eventualities as continuing and indefinite in temporal extent. We show that distributional semantic can reliably predict aspectual classes across languages, and achieves remarkable performance even in zero-shot cross-lingual experiments.

Our study has also revealed that these qualitative properties and grammatical differences reflect the discourse constraints in play when subjects write captions for images and that these findings are generalizable across languages. We have leveraged this observation for our computational work where we show that aspect can be predicted with distributional representations in a mono-lingual setup. We have furthermore provided first evidence that aspect can be predicted in a zero-shot cross-lingual manner where a model has not been exposed to any training data in the target language at all.

Acknowledgement

We would like to thank Aaron White, Gabriel Greenberg and the anonymous reviewers for their helpful comments. The research presented here is supported by NSF Awards IIS-1526723 and CCF-1934924.

References

- Mustafa Aksan and Yeşim Aksan. 2006. Denominal verbs and their aspectual properties. *Dil Dergisi*, pages 7 – 27.
- Yeşim Aksan. 2003. [Türkçe’de durum değişikliği eylemlerinin kılınıp özellikleri](#). *DİLBİLİM ARAŞTIRMALARI DERGİSİ*.
- Malihe Alikhani, Sreyasi Nag Chowdhury, Gerard de Melo, and Matthew Stone. 2019. Cite: A corpus of image-text discourse relations. *arXiv preprint arXiv:1904.06286*.
- Malihe Alikhani, Piyush Sharma, Shengjie Li, Radu Soricut, and Matthew Stone. 2020. [Cross-modal coherence modeling for caption generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6525–6535, Online. Association for Computational Linguistics.
- Malihe Alikhani and Matthew Stone. 2018. Exploring coherence in visual explanations. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 272–277. IEEE.
- Malihe Alikhani and Matthew Stone. 2019. “caption” as a coherence relation: Evidence and implications. In *Proceedings of the Second Workshop on Shortcomings in Vision and Language*, pages 58–67.
- Daniela Baiamonte, Tommaso Caselli, and Irina Prodanof. 2016. Annotating content zones in news articles. *CLiC it*, page 40.
- Raffaella Bernardi, Ruket Cakici, Desmond Elliott, Aykut Erdem, Erkut Erdem, Nazli Ikizler-Cinbis, Frank Keller, Adrian Muscat, and Barbara Plank. 2016. Automatic description generation from images: A survey of models, datasets, and evaluation measures. *Journal of Artificial Intelligence Research*, 55:409–442.
- Alan B Cantor. 1996. Sample-size calculations for cohen’s kappa. *Psychological methods*, 1(2):150.
- Tommaso Caselli and Valeria Quochi. 2007. Inferring the semantics of temporal prepositions in italian. In *Proceedings of the Fourth ACL-SIGSEM Workshop on Prepositions*, pages 38–44. Association for Computational Linguistics.
- Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. [Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation](#). In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64, Brussels, Belgium. Association for Computational Linguistics.
- Neil Cohn. 2013. Visual narrative structure. *Cognitive science*, 37(3):413–452.
- Bernard Comrie. 1976. *Aspect: An introduction to the study of verbal aspect and related problems*, volume 2. Cambridge university press.
- Samuel Cumming, Gabriel Greenberg, and Rory Kelly. 2017. Conventions of viewpoint coherence in film. *Philosophers’ Imprint*, 17(1):1–29.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [Bert: Pre-training of deep bidirectional transformers for language understanding](#).
- Bonnie J. Dorr and Mari Broman Olsen. 1997. Deriving verbal and compositional lexical aspect for nlp applications. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 151–158, Madrid, Spain. Association for Computational Linguistics.
- F Faul, E Erdfelder, AG Lang, and A Buchner. 2014. G* power: statistical power analyses for windows and mac.
- Hana Filip. 2004. The telicity parameter revisited. In *Semantics and Linguistic Theory*, volume 14, pages 92–109.
- George Forman and Martin Scholz. 2010. [Apples-to-apples in cross-validation studies: Pitfalls in classifier performance measurement](#). *SIGKDD Explor. Newsl.*, 12(1):49–57.
- Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.
- Jacqueline Guéron. 2008. On the difference between telicity and perfectivity. *Lingua*, 118(11):1816–1840.
- Liane Guillou, Sander Bijl de Vroe, Mohammad Javad Hosseini, Mark Johnson, and Mark Steedman. 2020. [Incorporating temporal information in entailment graph mining](#). In *Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs)*, pages 60–71, Barcelona, Spain (Online). Association for Computational Linguistics.
- Nizar Y. Habash. 2010. [Introduction to arabic natural language processing](#). *Synthesis Lectures on Human Language Technologies*, 3(1):1–187.
- Friedrich Hamm and Oliver Bott. 2018. Tense and Aspect. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, fall 2018 edition. Metaphysics Research Lab, Stanford University.
- Mainak Jas and Devi Parikh. 2015. Image specificity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2727–2736.

- Thomas Kober, Malihe Alikhani, Matthew Stone, and Mark Steedman. 2020. [Aspectuality across genre: A distributional semantics approach](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4546–4562, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Thomas Kober, Sander Bijl de Vroe, and Mark Steedman. 2019. [Temporal and aspectual entailment](#). In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 103–119, Gothenburg, Sweden. Association for Computational Linguistics.
- Mikhail Korobov. 2015. [Morphological analyzer and generator for russian and ukrainian languages](#). In Mikhail Yu. Khachay, Natalia Konstantinova, Alexander Panchenko, Dmitry I. Ignatov, and Valeri G. Labunets, editors, *Analysis of Images, Social Networks and Texts*, volume 542 of *Communications in Computer and Information Science*, pages 320–332. Springer International Publishing.
- Oksana Laleko. 2008. Compositional telicity and heritage russian aspect. In *Proceedings of the Thirty-Eighth Western Conference on Linguistics (WECOL)*, volume 19, pages 150–160.
- Sheng Li, Zhiqiang Tao, Kang Li, and Yun Fu. 2019. Visual to text: Survey of image and video captioning. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 3(4):297–312.
- Jo-Wang Lin. 2006. Time in a language without tense: The case of chinese. *Journal of Semantics*, 23(1):1–53.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.
- Maryam Majidi. 2011. Lexical aspect in farsi. In *Proceedings of the journal of Persian Literature*, pages 145–158.
- Scott McCloud. 1993. *Understanding comics: The invisible art*. William Morrow.
- Karine Megerdoomian. 2002. Aspect in complex predicates. In *Talk presented at the Workshop on Complex Predicates, Particles and Subevents, Konstanz*.
- Eleni Metheniti, Tim Van De Cruys, and Nabil Hathout. 2022. [About time: Do transformers learn temporal verbal aspect?](#) In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 88–101, Dublin, Ireland. Association for Computational Linguistics.
- Paramita Mirza and Sara Tonelli. 2014. [An analysis of causality between events and its relation to temporal information](#). In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2097–2106, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.
- Marc Moens and Mark Steedman. 1988. [Temporal ontology and temporal reference](#). *Computational Linguistics*, 14(2):15–28.
- Christoph Molnar. 2022. *Interpretable Machine Learning*, 2 edition.
- Kemal Oflazer, Elvan Göçmen, Elvan Gocmen, and Cem Bozsahin. 1994. [An outline of turkish morphology](#).
- Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholly, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. [MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 1094–1101, Reykjavik, Iceland. European Languages Resources Association (ELRA).
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. [Scikit-learn: Machine learning in python](#). *Journal of Machine Learning Research*, 12:2825–2830.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [Deep contextualized word representations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.
- James Pustejovsky, Kiyong Lee, Harry Bunt, and Laurent Romary. 2010. Iso-timeml: An international standard for semantic annotation. In *LREC*, volume 10, pages 394–397.
- Karin Kipper Schuler and Martha S. Palmer. 2005. *Verbnet: A Broad-Coverage, Comprehensive Verb Lexicon*. Ph.D. thesis, USA. AAI3179808.
- L. S. Shapley. 1953. *A Value for n-Person Games*, pages 307–317. Princeton University Press.
- Dan I Slobin. 2004. The many ways to search for a frog. *Relating events in narrative*, 2:219–257.
- Andrew Spencer. 1991. *Morphological theory: An introduction to word structure in generative grammar*. Wiley-Blackwell.

- Milan Straka and Jana Straková. 2017. [Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe](#). In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics.
- Junyi Sun. 2012. Jieba. *Chinese word segmentation tool*.
- Emiel van Miltenburg, Ákos Kádár, Ruud Koolen, and Emiel Krahmer. 2018a. [DIDEC: The Dutch image description and eye-tracking corpus](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3658–3669, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Emiel van Miltenburg, Ruud Koolen, and Emiel Krahmer. 2018b. [Varying image description tasks: spoken versus written descriptions](#). In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 88–100, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.
- Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.
- Serkan Çakmak. 2013. "var" ve "yok" sözcüklerinin morfolojik kimliği. *International Periodical For The Languages, Literature and History of Turkish or Turkic*, 8(4):463–471.

7 Supplemental Material

	Wikipedia	Caption
Arabic	11.60	4.65
Chinese	21.13	10.63
Farsi	24	7
German	13.43	9.47
Russian	15.43	4.27
Turkish	12.76	10.90

Table 4: Wikipedia sentences are on average longer, i.e. contain more tokens, than captions.

Aspect		Arabic		Chinese		Farsi		German		Russian		Turkish	
		Capt.	Wiki	Capt.	Wiki	Capt.	Wiki	Capt.	Wiki	Capt.	Wiki	Capt.	Wiki
fastText	Atelic	0.84	-	0.12	-	0.67	-	0.83	-	0.80	-	0.45	-
	Telic	-	0.67	-	0.62	-	0.28	-	0.79	-	0.62	-	0.65
	State	0.10	0.66	0.53	0.89	0.46	0.59	0.76	0.71	0.27	0.84	0.42	0.53
mBERT	Atelic	0.88	-	0.59	-	0.88	-	0.48	-	0.78	-	0.44	-
	Telic	-	0.63	-	0.79	-	0.69	-	0.68	-	0.84	-	0.70
	State	0.21	0.76	0.85	0.41	0.62	0.61	0.66	0.78	0.17	0.50	0.65	0.84
ELMo	Atelic	0.85	-	0.00	-	0.45	-	0.79	-	0.17	-	0.38	-
	Telic	-	0.00	-	0.54	-	0.40	-	0.27	-	0.29	-	0.82
	State	0.10	0.74	0.81	0.00	0.35	0.46	0.00	0.55	0.56	0.55	0.48	0.00

Table 5: Zero-shot cross-lingual F1-scores per label across all languages with using fastText embeddings (top), multilingual BERT embeddings (middle) and ELMo embeddings (bottom).

Nominal Metaphor Generation with Multitask Learning

Yucheng Li¹, Chenghua Lin², Frank Guerin¹

¹ Department of Computer Science, University of Surrey, UK

{yucheng.li, f.guerin}@surrey.ac.uk

² Department of Computer Science, University of Sheffield, UK

c.lin@sheffield.ac.uk

Abstract

Nominal metaphors are frequently used in human language and have been shown to be effective in persuading, expressing emotion, and stimulating interest. This paper tackles the problem of Chinese Nominal Metaphor (NM) generation. We introduce a novel multi-task framework, which jointly optimizes three tasks: NM identification, NM component identification, and NM generation. The metaphor identification module is able to perform a self-training procedure, which discovers novel metaphors from a large-scale unlabeled corpus for NM generation. The NM component identification module emphasizes components during training and conditions the generation on these NM components for more coherent results. To train the NM identification and component identification modules, we construct an annotated corpus consisting of 6.3k sentences that contain diverse metaphorical patterns. Automatic metrics show that our method can produce diverse metaphors with good readability, where 92% of them are novel metaphorical comparisons. Human evaluation shows our model significantly outperforms baselines on consistency and creativity.

1 Introduction

Metaphors are commonly used in human language. Usually, metaphors compare two different kinds of objects or concepts with the intent to make the expression more vivid, or to make unfamiliar things easier to understand (Paul, 1970). According to contrastive studies of English and Chinese, metaphors are especially crucial in Chinese as there are fewer abstract words in Chinese, so that people tend to express abstract meaning via metaphors (Lian, 1994).

In this paper, we focus on the generation task of a special type of Chinese metaphor – Nominal Metaphors (NMs). NMs (比喻 in Chinese) are figures of speech associating a noun with another noun through a COMPARATOR such as *like*,

1. 这个[孩子] _{tenor} 壮的像[牛] _{vehicle} This [boy] _{tenor} is as strong as a [bull] _{vehicle} .	<i>Nominal</i>
2. [生活] _{tenor} 好比[旅行] _{vehicle} , 没有计划就难以前行 [Life] _{tenor} is a [journey] _{vehicle} , we cannot move on without a plan.	<i>Nominal</i>
3. Meta股价[跳水] _{metaphorical} META stock price [dives] _{metaphorical} .	<i>Verbal</i>
4. 他可以像大厨一样烹饪 He can cook like a pro.	<i>Literal</i>

Table 1: Examples of Chinese nominal metaphor, verbal metaphor, and NM components. Note that when the words “like” or “as” are used as COMPARATORS, we also call these special NMs 明喻 (Similes).

be, become in English and 像,是,变成 in Chinese. Examples and NM components are shown in Table 1. In addition to the COMPARATOR (**bold**) there are three other components in a nominal metaphor: TENOR, VEHICLE, and CONTEXT (text with underline). The TENOR is the subject of the metaphor, and the VEHICLE is the source of the imagery (i.e., the object of metaphor). CONTEXT is used to explain the comparison and is crucial for understanding the comparison (more details about NM and NM components in § 2.1). The NM generation task is as follows: given a TENOR, generate a metaphor containing the three remaining NM components, i.e., VEHICLE, COMPARATOR and CONTEXT. Previous efforts on NM processing mainly engage in identification (Liu et al., 2018; Zeng et al., 2020) and interpretation (Su et al., 2016, 2017), generation of NMs has not been well studied, despite the benefits it can bring to many downstream tasks. Glucksberg (1989); Zhou (2020) suggest that metaphors are important to an engaging conversation and can effectively stimulate user interest in communicating with chatbots. Chakrabarty et al. (2020, 2021) show that users

prefer stories and poems enhanced with metaphor generation by replacing literal expressions with generated metaphors.

To tackle the Chinese NM generation, there are mainly two challenges to address. First, existing Chinese NM corpora are not large enough to power current data-driven text generation approaches. Second, the auto-regressive nature of generative models always assigns higher priority to fluency, which makes the metaphor generation procedure produce *inconsistency errors* (i.e., generating nonsense comparisons without CONTEXT explaining)¹ and *literal errors* (i.e., generating literal expressions).

We propose a novel multitask approach for Chinese NM generation called MetaGen to address the above mentioned problems. Specifically, three tasks are jointly optimized: NM generation, NM identification, and NM components identification. **First**, for the data scarcity problem, we perform a self-training procedure to learn newly discovered metaphors from large-scale unlabeled datasets. Self-training has three main steps: 1) our model is trained on a labeled dataset for NM identification; 2) we apply our model on an unlabeled corpus to detect potential NMs with a corresponding confidence score; and 3) train an NM generation model on the combination of labeled and newly found NMs. By exploiting rich metaphors from large-scale resources, the performance of MetaGen can be significantly improved yet the data requirement can be dramatically reduced. **Second**, MetaGen proposes to identify potential metaphor components (i.e., TENOR, COMPARATOR and VEHICLE) supervised by the attention weights generated by the NM classifier. To alleviate *inconsistency errors*, MetaGen conditions the generation process on the potential NM components; this enforces the CONTEXT generation to depend on the comparison, rather than producing fluent but bland CONTEXT that does not explain the comparison. In terms of the *literal errors*, NMs components are emphasised via attention weight to encourage MetaGen produce metaphorical expressions rather than literals.

We also build an annotated corpus for Chinese NM identification consisting 6.3k sentences. Instead of focusing on a specific metaphorical pattern (Liu et al., 2018), our corpus con-

¹An example of *inconsistency error*: “Teacher is like a candle, floating gently in the air”. Although the comparison is valid, the CONTEXT is inconsistent with the comparison. This also shows the importance of CONTEXT in NM generation.

tains diverse nominal metaphorical usages. We also ensure the CONTEXT is explicit for each metaphor annotated, and the TENOR of each metaphor is also identified. Source code and data can be found in https://github.com/liyucheng09/Metaphor_Generator.

2 Related Work

2.1 Metaphors in Chinese

Following (Krishnakumaran and Zhu, 2007; Rai and Chakraverty, 2020), we can divide English metaphors into four types as follows:

Type-I: (Nominal Metaphors) A noun is associated with another noun through the comparators, e.g., “Love is a journey”.

Type-II: (Verbal Metaphors or Subject-Verb-Object (SVO) metaphors) Sentences with metaphorical verb, e.g., “He kills a process”.

Type-III: (Adjective-Noun (AN) metaphor) Metaphorical adjectives with a noun fall into this category, e.g., “sweet boy”.

Type-IV: (Adverb-Verb (AV) metaphor) Metaphorical adverbs with a verb, e.g., “speak fluidly”.

However, the definition of metaphor in the context of Chinese is slightly different from its English counterpart (Wang, 2004). 比喻 (Metaphor), or 打比方 (draw an analogy), which draws a comparison between objects or concepts, mainly means *Type-I* metaphor, i.e., NMs. A specific term 比拟 (Personification/Match) is used to indicate *Type-II*, *Type-III*, *Type-IV* metaphors in Chinese, which aims to describe an object or concept in a view of a person or another object. Verbal Metaphors (VMs) are the most frequent type of metaphor in English (Martin, 2006; Steen, 2010), but NMs are the dominant figurative language in Chinese. According to a small scale annotation analysis (Su et al., 2016), NMs are around four times more frequent than VMs in Chinese. Lian (1994) gives a possible explanation for this phenomenon: Chinese people tend to express abstract concepts via nominal metaphors or idioms as there are fewer abstract terms in Chinese than in English. For example, a Chinese nominal metaphor “像竹篮打水” (doing something is like ladling water to a leaky basket), is used to express the meaning of “hopeless”.

Chinese NMs often consist of four components: TENOR, VEHICLE, COMPARATOR, (本体, 喻体, 比喻词 in Chinese) and CONTEXT, as shown in table 1. The CONTEXT here is a component used

to *explain* the comparison; its definition is relatively flexible. Sometimes it can be a simple adjective, sometimes a relative clause, or even implicit in some cases. For example, the NM “The city is like a painting” omits the textual CONTEXT to emphasize visual senses. However, CONTEXT is extremely important in helping readers to understand the comparison. According to Indurkha (2007) and Lakoff and Johnson (2008), a comparison can be drawn between any concepts, but it must have a CONTEXT to explain the comparison or to make the comparison coherent to daily experience. Considering the importance of CONTEXT, we **do not** consider a comparison without CONTEXT as a successfully generated NM case in our experiments. Additionally, there are two linguistic principles Chinese NMs must obey (Wang, 2004): 1) The comparison must be drawn between two concepts with different natures; and 2) the two concepts being compared should share commonalities. Specifically, the COMPARATOR “like” in the example No.4 does not necessarily make it an NM, since the comparison is drawn between the same concept “me cooking” and “pro cooking”. The second principle also emphasises the importance of CONTEXT. In summary, even though NMs usually share a relatively simple structure, Chinese NM generation is still challenging due to the requirement of providing CONTEXT and the necessity of understanding the relation between TENOR and VEHICLE.

2.2 Computational Processing of NMs

Previous works on computational processing of NMs can be classified into detection, interpretation and generation.

Detection and Interpretation Krishnakumaran and Zhu (2007) exploit the absence of a hyponymy relation between subject and object to identify metaphorical utterances. Shlomo and Last (2015) propose a random forest-based classifier for NM identification using both conceptual features such as abstractness and semantic relatedness such as domain corpus frequency. Su et al. (2016) follow the idea of lack of hyponymy relationship from (Krishnakumaran and Zhu, 2007) and realize it using cosine distance between pre-trained word2vec embeddings of the source and target concepts. Liu et al. (2018); Zeng et al. (2020) tackle Chinese simile detection by designing a multi-task framework and a local attention mechanism. Su et al. (2016, 2017) focus on NM interpretation and per-

form experiments on both English and Chinese NMs. They extract properties of TENOR and VEHICLE from WordNet and use pre-trained word2vec embeddings to identify related properties shared by both components.

Generation Despite the benefits NM generation can bring to the community, prior works on this task are relative sparse. Early works often rely on templates. Terai and Nakagawa (2010) compute the relatedness between concepts with computational language analysis and select candidates to fill metaphor templates like “A is like B”. Veale (2016) uses a knowledge-base to generate *XYZ* style NMs such as “Bruce Wayne is the Donald Trump of Gotham City”. Zhou (2020) not only choose candidate concept pairs by word embedding similarity to fill the template but also choose appropriate COMPARATORS to link the concept pair. (Chakrabarty et al., 2020) introduce a neural style transfer approach for simile generation, which fine-tunes a pre-trained sequence-to-sequence model on a literal-simile parallel dataset. Nevertheless, previous template-based approaches heavily constrain the diversity of generated NMs and both template methods and neural methods produce NMs in a relative simple structure. Most importantly, previous methods do not provide CONTEXT in their generations (or only provide little CONTEXT), which makes generated results less readable.

3 Method

Given an object or concept as a starting TENOR, a Chinese nominal metaphor will be generated consisting of four NM components: a comparison between TENOR and VEHICLE linked with a COMPARATOR and a CONTEXT as an explanation for the comparison. The overall multitask framework is shown in Figure 1. We can roughly divide our framework into four elements: 1) the GPT2 (Radford et al., 2019) pre-trained language model; and three task-specific fully-connected layers used for 2) NM identification; 3) NM components identification; and 4) NM generation.

3.1 Shared Representation

Since we are tackling a generation task, we employ a pre-trained unidirectional transformer-based language model, GPT2, as our basic encoder. Contextualized words’ representations are obtained after feeding words to the GPT2 model. Formally, given sentence $S = (w_0, \dots, w_n, w_{EOS})$,

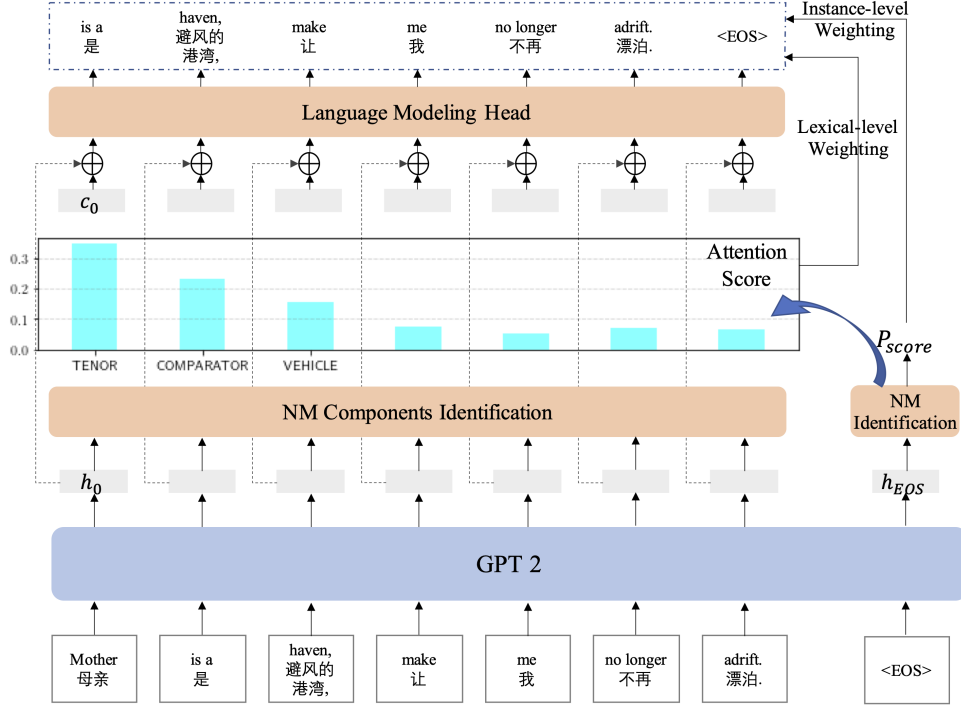


Figure 1: The overall framework.

the GPT2 model produces a list of vectors $H = (h_0, \dots, h_n, h_{EOS})$, where the EOS is a special delimiter indicating the end of the sentence. Note that the representation here are used in the three individual tasks described below and the parameters are also shared across all tasks.

3.2 Task 1: NM Identification

The NM identification module is used to assign metaphorical probability to sentences. This score will be used in the Self-Training procedure (described in §3.4). Specifically, we use h_{EOS} as the sentence representation of S (similar to the usage of `cls` embedding in BERT-based systems (Devlin et al., 2018)) and apply a linear layer plus a softmax layer on it to compute the metaphorical probability of the sentence S . Formally, the metaphor probability is computed as follow:

$$P_M = \text{softmax}(W_m h_{EOS} + b_m) \quad (1)$$

where W_m and b_m is a trainable weight and bias for NMs identification.

We train this module on a supervised dataset noted as $U = \{(x_i, y_i)\}_{i=1}^N$, where x indicates the sentence and y indicates whether x is a NM. In summary, we minimize the following loss function for NM identification:

$$L_1 = - \sum_{x \in U} \log P(\hat{y}|x) \quad (2)$$

3.3 Task 2: NMs Components Identification

Although GPT2 model is powerful in generating fluent and grammatical text, it still suffers from incoherence issues (Ko and Li, 2020; Tan et al., 2021). In the scenario of NM generation, it means the CONTEXT generation might be inconsistent with the metaphorical comparison thus resulting in *inconsistency errors*. Besides, the innate tendency of generative models to produce literal text often leads to *literal errors* (Chakrabarty et al., 2021).

To address the *inconsistency errors*, our model conditions the generation procedure on the metaphorical comparison, that is the NM components TENOR, VEHICLE, and COMPARATOR. We also weight these NM components with higher score during training to alleviate *literal errors*. These two approaches (described in §3.4) require the involvement of NM components, therefore, we apply a linear layer to compute the probability for each token to be an NM component. Formally, this probability is computed as follows:

$$P_c = \text{Sigmoid}(W_c H + b_c) \quad (3)$$

where W_c and b_c are trainable weights and bias for NM component identification, and P_c is the resulting probability distribution. Note that this process does not predict the type of components (e.g., TENOR), instead, it only computes a proba-

bility for each token indicating the extent to which the generation should focus on each.

We propose to use the attention weights generated from the NM classifier (obtained in §3.2) as the supervision signals for NM component identification. As shown in (Liu et al., 2018; Zeng et al., 2020), the metaphor classifier tends to focus more on corresponding metaphor components, we thus use this property to discover NM components. Specifically, we use KL divergence to have our distribution P_c as close as possible to the attention weights Φ .

$$L_2 = D_{KL}(P_c \parallel \Phi) \quad (4)$$

where Φ is the self-attention score the h_{EOS} attending to other tokens generated by the last layer Transformer of GPT2.

$$\Phi = \text{softmax}\left(\frac{Qk^T}{\sqrt{d_k}}\right) \quad (5)$$

The Q here is the Query matrix for self-attention, and k is the Value vector only for the EOS token.

3.4 Task 3: NM Generation

We perform the NM generation task with three steps: 1) conditioning the generation on NM components; 2) emphasizing the NM components; and 3) executing the self-training procedure.

Conditioning To allow token predictions conditioned on NM components, we provide a list of NM component representations $C = (c_0, \dots, c_i, \dots, c_n)$ for each prediction step respectively. Then the NM component representation c_i is fed into the language modeling head together with the contextualized token embedding h_i . Formally, c_i is computed as follows:

$$c_i = \sum_{k=0}^i \alpha_k \cdot h_k \quad (6)$$

where the weight score α is computed as follows:

$$(\alpha_0, \dots, \alpha_i) = \text{softmax } P_c^{\{0, \dots, i\}} \quad (7)$$

The c_i here mainly captures NM component information before the i -th token (i.e., NM components within (w_0, \dots, w_i)). Then we concatenate the contextualized token embedding h_i and its corresponding NM component information embedding c_i to predict the next token.

$$P(w_{i+1}|w_0, \dots, w_i) = \text{softmax} [W_l(h_i \oplus c_i) + b_l] \quad (8)$$

where the W_l and b_l are trainable weight matrix and bias, \oplus indicating the concatenation operation.

Emphasizing We emphasize the NM components during training by directly applying attention weight P_c on the loss function. Specifically, given a sentence $S = (w_0, \dots, w_n)$, we minimize the following loss function:

$$\mathcal{L}(S) = - \sum_{i=0}^n P_c^i \cdot \log P(w_i|w_0, \dots, w_{i-1}) \quad (9)$$

where P_c^i is the probability to be one of the NM components of token w_i .

Self-training Self-training is an effective approach to tackle data scarcity and has been successfully used in many downstream tasks (He et al., 2019; Parthasarathi and Strom, 2019; Xie et al., 2020). In our setting, we adopt self-training for discovering novel Chinese NMs from large-scale corpora to train the NM generation module so that the fluency and diversity of generation can be improved.

Formally, given an unlabeled corpus $V = \{x_i\}_{i=0}^N$ where each x is a sentence $x = (w_0, \dots, w_n)$, the NM identifier will assign a probability to each x_i noted as P_M^i . We then mix the unlabeled corpus $V = \{(x_i, P_M^i)\}_{i=0}^N$ and supervised dataset $U = \{(x_i, y_i)\}_{i=1}^N$ together, and train the overall framework on it. Formally, we minimize the following loss function:

$$L_3 = - \left[\sum_{x \in V} P_M^i \cdot \mathcal{L}(x) + \sum_{S \in U} \mathcal{L}(S) \right] \quad (10)$$

3.5 Training and Inference

The final loss function of our framework is a weighted sum of three task-specific loss function.

$$L = \gamma \cdot L_1 + L_2 + L_3 \quad (11)$$

Note that when learning unlabeled sentences, γ is set to 0, since these instances lack the supervision label for NM identification. To help our model converge, before training the overall framework on the mixed data by L , we pre-train our model on the supervised dataset for Task 1 first. Besides, when doing inference, our model only performs Task 3.

4 Experiment

4.1 dataset

To train our multitask framework, we construct two datasets: a supervised Chinese NM Corpus (CMC)

	CMC	CLC
# Sentences	6257	6.98M
# NM	2787	-
# literal sentence	3554	-
# tokens	225K	202M
# tokens per sentence	35	29

Table 2: Statistics of CMC and CLC datasets

and a large-scale unsupervised Chinese Literature Corpus (CLC).

CMC Existing Chinese metaphor corpus are neither too small, like [Su et al. \(2016\)](#) contains 120 examples, or focusing on a specific metaphorical pattern, like [Liu et al. \(2018\)](#) contains sentences with a specific COMPARATOR 像 (like). In our corpus, we try to include nominal metaphors as diverse as possible. The annotation of the CMC consists of four steps: 1) we collect 55,000 Chinese sentences from essays, articles, and novels; 2) we employ three Chinese graduate students with background of NLP to label each sentence as a NM or not; 3) we take the majority agreement as the final label for each sentence; 4) the boundary of TENOR is identified at last. To encourage the CONTEXT to be generated, we ensure CONTEXT occurs explicitly in each metaphor we labeled. Before the annotation, annotators are trained with examples and instructed with basic Chinese NM principles (described in §2.1). We compute the inner-annotator agreement of NM label via Krippendorff’s alpha ([Krishnakumar and Zhu, 2007](#)). The agreement rate is 0.84. Statistics of CMC are shown in Table 2. Some examples are shown in Appendix A.1.

CLC In self-training, we need a large-scale corpus so that the NM identifier can discover novel NMs. However, popular Chinese corpora, such as news, Wikipedia, web pages, are not suited to be used as metaphor resources. Intuitively, literature text might be a promising resource of diverse metaphors. Therefore, we construct a Chinese literature corpus by collecting a large number of essays, novels, and fictions (see details in Appendix A.2). Statistics of CLC are shown in Table 2.

4.2 Baselines

Chinese NMs generation is a novel task, we select three general generative models and an English simile generation method as baselines.

SeqGAN: Sequence Generative adversarial network ([Yu et al., 2017](#)) with a generator imple-

mented by LSTM network and a discriminator implemented by CNN network. We train this model on CMC to produce Chinese NM.

GPT2: The Chinese GPT2 model is fine-tuned on the CMC dataset to produce Chinese NMs as a baseline model.

BART: We fine-tune a Chinese version BART model ([Shao et al., 2021](#)) model on parallel data pairs <TENOR, Sentence> obtained from CMC.

SCOPE: ([Chakrabarty et al., 2020](#)) SOTA method on English simile generation tasks, which fine-tunes BART model on a large-scale automatically created literal-simile parallel corpus.

4.3 Experiments Setting

We use a pre-trained Chinese GPT2 model² to avoid starting training from scratch. Our model is pre-trained on NM identification task with CMC for 3 epochs before jointly optimizing three task-specific loss functions. The implementation of SeqGAN³ and the pre-trained Chinese BART model⁴ can be found in the footnote. Before the SeqGAN starts training on CMC, we first pre-train the generator of SeqGAN on CLC for 50k steps. Hyperparameters not specified are all followed by default settings. Note that the SCOPE model is designed for English Simile generation and it takes a literal utterance as input. To compare SCOPE results with our method, we first translate input TENORS into English (via Google Translator), then translate generated NMs back to Chinese (details in Appendix B). In the test stage, we randomly select and feed 200 TENORS from CMC to all generative models. During decoding, all beam sizes are set to 12, thus each model generated 12 sentence for each TENOR. In total, 2400 sentences are obtained per model for testing.

4.4 Metrics

Automatic Metrics We use perplexity (PPL) to evaluate the fluency of the generated text, which is calculated by an open source Chinese language model ([Zhang et al., 2020](#)). **Dist-1, Dist-2** ([Li et al., 2016](#)) compute the distinct unigrams and bigrams ratio of generated text which are used to measure model’s ability to produce diversity outputs. To test the **metaphoricity (Meta)** of generated outputs, we

²<https://huggingface.co/uer/gpt2-chinese-cluecorpus-small>

³<https://github.com/LantaoYu/SeqGAN>

⁴<https://huggingface.co/fnlp/bart-base-chinese>

Methods	PPL	Dist-1	Dist-2	Meta	Novelty	Fluency	Consistency	Creativity
SeqGAN	89.43	.00336	.0116	.998	.200	3.33 (.51)	3.80 (.46)	1.67 (.34)
GPT2	57.88	.00916	.1154	.981	.800	4.00 (.62)	3.10 (.39)	2.60 (.31)
BART	48.58	.00826	.0971	.978	.725	4.35 (.54)	3.05 (.37)	2.30 (.32)
SCOPE	92.32	.00517	.0673	.910	.385	3.10 (.64)	2.70 (.44)	2.10 (.45)
Our Method	25.79	.01153	.1674	.948	.920	4.65 (.58)	4.40 (.45)	3.80 (.36)
w/o Self-training	62.54	.00674	.0906	.982	.785	3.85 (.54)	3.87 (.42)	2.76 (.38)
w/o Emphasizing	25.58	.01150	.1529	.803	.900	4.50 (.63)	3.91 (.32)	3.41 (.43)
w/o Conditioning	24.93	.01053	.1534	.875	.930	4.25 (.61)	3.05 (.45)	3.24 (.39)

Table 3: Results of automatic metrics and human evaluation. Boldface denotes the best results among our method and baselines. The inter-annotator agreement for human evaluation are shown in parenthesis.

train a RoBERTa-based Chinese NM classifier on CMC to compute the ratio of metaphorical utterances in the generated sentences. The accuracy of this classifier is 97.89%, which is reasonable enough to perform evaluation (details in Appendix C). **Novelty** is to test how well models can generate metaphors they have never seen during training. We use a syntax-based approach to identify TENORS and VEHICLES from generated NMs and compute the proportion of <TENOR, VEHICLE> pair that does not co-occur in the training set.

Human Evaluation Due to the creative and delicate usage of NM, automatic metrics are not adequate to test the quality of generated outputs. We also perform human evaluation based on the following three criteria: 1) **Fluency** indicates how well the metaphor is formed; whether the expression is grammatical and fluent. 2) **Consistency** indicates whether the metaphor can explain itself; how well the VEHICLE relate to TENOR and how well the CONTEXT explain the comparison. 3) **Creativity** scores how creative annotators think the metaphor is. Note that the Creativity judgment is based on annotators’ real-life experience, rather than measuring whether the generated metaphor appears in the training dataset. Three annotators were instructed to rate the three criteria from 1 to 5 where 1 denotes worst and 5 be the best.

5 Results

5.1 Automatic Evaluation

Results of automatic metrics are shown in Table 3. Our method significantly outperforms baselines in most automatic metrics. Our model obtains a lower PPL, which illustrates our model is better at producing fluency and grammatical text. Higher Dist-1 and Dist-2 scores show our method produces less repetitive unigrams and bigrams during generation,

which is essential in creative language generation. The Meta (metaphor) score shows that our model produces more literal expressions than baselines, which might result from the self-training procedure, where non-metaphorical sentences are sometimes wrongly identified by the NM identification module, and thus there is noise in NM modeling. The highest Novelty score demonstrates our method’s ability to generate creative comparisons.

We implemented an ablation study to test the effectiveness of self-training, NM component emphasizing, and context conditioning. Experimental results prove the self-training mechanism improves both generation fluency and diversity. Removing self-training from our model affects four automatic metrics by a large margin. The NM component emphasizing mainly helps our method alleviate *literal errors* and thus improve the Meta score. The context conditioning also benefits the overall framework in Meta score.

5.2 Human Evaluation

We select 180 sentences in total to annotate (15 TENORS, 12 sentences for each TENOR). Human evaluation results are shown in Table 3. The Table also shows the inter-annotator agreement of human annotation via Krippendorff’s alpha. We can see that our method beats four baseline models on all three human-centric metrics. The most significant improvement lies in Consistency and Creativity, which show our method can not only generate creative comparisons, but, most importantly, also provide a CONTEXT for each NM to explain the comparison, which is essential for readability. Human evaluation also demonstrates the effectiveness of self-training, emphasizing, and conditioning. Self-training enhances generation quality in both fluency and creativity dimensions. Conditioning mostly contributes to the consistency score.

Methods	Text (Chinese)	Text (Translated)	Con.	Cre.
GPT2	秋天是美丽的，让人赏心悦目。	Autumn is beautiful, and is delightful to the eye.	-	-
	秋天是个动情的音符，荡漾在夏日的清纯中。	Autumn is an emotional note, rippling in the purity of summer.	2.0	3.5
	秋天是最好的伴奏曲，让世界充满微笑。	Autumn is the best concertos, making the world full of smiles.	3.3	2.0
SCOPE	秋天象征春天，像一个月前。	Autumn is a symbol of spring, like a month ago.	-	-
	秋天象征热情，像一个情人。	Autumn is a symbol of passion, like a lover.	3.3	1.3
	秋天象征爱情，像一个女人。	Autumn is a symbol of love, like a woman.	1.3	2.7
Our method	秋天像一只彩笔画般的画笔，勾勒出一幅幅多彩多姿的画卷。	Autumn is like a multi-colored paintbrush, sketching out colorful pictures.	5.0	2.7
	秋天像小姑娘的脚，带着她那柔软的臂膀，在枝头翩翩起舞。	Autumn is like a little girl’s feet with her softness. Arms, dancing in the branches.	3.7	5.0
	秋天像刚刚落地的苹果，在果园里露出个头。	Autumn is like an apple that has just fallen, showing its head in the orchard.	4.3	4.0
	秋天像刚落蝉的蝉，婉转地鸣叫着，见证着树梢上金黄色的叶子慢慢向蓝天生长。	Autumn is like a cicada that has just fallen, chirping tactfully, seeing the golden leaves on the treetops grow towards the blue sky slowly.	4.3	5.0

Table 4: NMs generated by our method and baselines given a TENOR 秋天 (Autumn). **Con.** and **Cre.** indicate the two human evaluation metrics **Consistency** and **Creativity** respectively. We do not assign Con. and Cre. score for non-metaphorical utterances. More examples of MetaGen are shown in Appendix D.

5.3 Case Study

Generated examples of GPT2, SCOPE, and our model are shown in Table 4. The corresponding Consistency and Creativity score are also given. In this table, models generate NMs by taking 秋天 (Autumn) as the input TENOR. We see that although all three models are able to produce metaphorical outputs, the quality of generated results differs among systems. **First**, the comparisons given by our model are more diverse than baselines. We can identify similar patterns in the outputs of GPT2 and SCOPE. For example, GPT2 tends to compare autumn with “music” (i.e., note and accompaniment) and SCOPE is likely to relate autumn with love (i.e., lover and woman). **Second**, CONTEXT produced by our method can explain the comparison well, which ensures the consistency and readability of the outputs. However, baselines are either give little CONTEXT (like SCOPE gives an adjective or noun as CONTEXT) or inappropriate CONTEXT (like GPT2 uses summer in the comparison of autumn). **Third**, we find our method

generates NMs in a relatively more complicated structure and speaks in a more poetic way. For example, our method does not use a single word as VEHICLE, instead, it generates detailed phrases, such as “apple that has just fallen”, “dancing on the branches”. These detailed components paint a more vivid picture, and thus improve the overall readability. The corresponding human-rated Consistency and Creativity scores support this.

6 Conclusion

In this paper, we introduce a novel language generation task: Chinese nominal metaphor generation. We also propose a multitask framework for Chinese nominal metaphor generation. Additionally, we publish an annotated corpus for Chinese nominal metaphors. Future directions can be trying the usage of syntactic features and controllable NM generation. Moreover, we would also like to evaluate the effect of metaphor generation in downstream tasks, such as story generation, dialog systems, and educational scenarios.

References

- Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779.
- Tuhin Chakrabarty, Smaranda Muresan, and Nanyun Peng. 2020. Generating similes effortlessly like a pro: A style transfer approach for simile generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6455–6469.
- Tuhin Chakrabarty, Xurui Zhang, Smaranda Muresan, and Nanyun Peng. 2021. Mermaid: Metaphor generation with symbolism and discriminative decoding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4250–4261.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Sam Glucksberg. 1989. Metaphors in conversation: How are they understood? why are they used? *Metaphor and Symbol*, 4(3):125–143.
- Junxian He, Jiatao Gu, Jiajun Shen, and Marc’Aurelio Ranzato. 2019. Revisiting self-training for neural sequence generation. In *International Conference on Learning Representations*.
- Bipin Indurkha. 2007. Creativity in interpreting poetic metaphors. In T. Kusumi, editor, *New directions in metaphor research*, pages 483–501. Tokyo, Japan: Hitsuji Shobo.
- Wei-Jen Ko and Junyi Jessy Li. 2020. Assessing discourse relations in language generation from gpt-2. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 52–59.
- Saisuresh Krishnakumaran and Xiaojin Zhu. 2007. Hunting elusive metaphors using lexical resources. In *Proceedings of the Workshop on Computational approaches to Figurative Language*, pages 13–20.
- George Lakoff and Mark Johnson. 2008. *Metaphors we live by*. University of Chicago press.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119.
- Shuneng Lian. 1994. *Contrastive Studies OF English and Chinese*. Fudan University Press.
- Lizhen Liu, Xiao Hu, Wei Song, Ruiji Fu, Ting Liu, and Guoping Hu. 2018. Neural multitask learning for simile recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1543–1553.
- James H Martin. 2006. A corpus-based analysis of context effects on metaphor comprehension.
- Sree Hari Krishnan Parthasarathi and Nikko Strom. 2019. Lessons from building acoustic models with a million hours of speech. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6670–6674. IEEE.
- Anthony M Paul. 1970. Figurative language. *Philosophy & Rhetoric*, pages 225–248.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Sunny Rai and Shampa Chakraverty. 2020. A survey on computational metaphor processing. *ACM Computing Surveys (CSUR)*, 53(2):1–37.
- Yunfan Shao, Zhichao Geng, Yitao Liu, Junqi Dai, Fei Yang, Li Zhe, Hujun Bao, and Xipeng Qiu. 2021. Cpt: A pre-trained unbalanced transformer for both chinese language understanding and generation. *arXiv preprint arXiv:2109.05729*.
- Yosef Ben Shlomo and Mark Last. 2015. Mil: automatic metaphor identification by statistical learning. In *Proceedings of the 2nd International Conference on Interactions between Data Mining and Natural Language Processing-Volume 1410*, pages 19–29.
- Gerard Steen. 2010. *A method for linguistic metaphor identification: From MIP to MIPVU*, volume 14. John Benjamins Publishing.
- Chang Su, Shuman Huang, and Yijiang Chen. 2017. Automatic detection and interpretation of nominal metaphor based on the theory of meaning. *Neurocomputing*, 219:300–311.
- Chang Su, Jia Tian, and Yijiang Chen. 2016. Latent semantic similarity based interpretation of chinese metaphors. *Engineering Applications of Artificial Intelligence*, 48:188–203.
- Bowen Tan, Zichao Yang, Maruan Al-Shedivat, Eric Xing, and Zhiting Hu. 2021. Progressive generation of long text with pretrained language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4313–4324.
- Asuka Terai and Masanori Nakagawa. 2010. A computational system of metaphor generation with evaluation mechanism. In *International Conference on Artificial Neural Networks*, pages 142–147. Springer.

Tony Veale. 2016. Round up the usual suspects: Knowledge-based metaphor generation. In *Proceedings of the Fourth Workshop on Metaphor in NLP*, pages 34–41.

Xijie Wang. 2004. *HanYu XiuCi Xue*. The Commercial Press.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. 2020. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Jiali Zeng, Linfeng Song, Jinsong Su, Jun Xie, Wei Song, and Jiebo Luo. 2020. Neural simile recognition with cyclic multitask learning and local attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9515–9522.

Zhengyan Zhang, Xu Han, Hao Zhou, Pei Ke, Yuxian Gu, Deming Ye, Yujia Qin, Yusheng Su, Haozhe Ji, Jian Guan, et al. 2020. Cpm: A large-scale generative chinese pre-trained language model. *arXiv preprint arXiv:2012.00413*.

Jin Zhou. 2020. Love is as complex as math: Metaphor generation system for social chatbot. In *Chinese Lexical Semantics: 20th Workshop, CLSW 2019, Beijing, China, June 28–30, 2019, Revised Selected Papers*, volume 11831, page 337. Springer Nature.

A Dataset

A.1 Chinese NM Corpus (CMC)

Examples in CMC are shown in Table 5.

A.2 Chinese Literature Corpus (CLC)

CLC consists of three main categories of Chinese literature: Children’s Literature (Children), Chinese Literature (Chinese), Translated Literature (Translated). Statistics of each category are shown in Table 6.

B SCOPE Model

SCOPE model takes a literal expression as input and produces a simile correspondingly. For example, given “the city is beautiful”, SCOPE model will transfer the literal expression into a simile: “The city is like a painting”.

In our experiments, to compare SCOPE with our method, we first 1) feed a TENOR to COMET (Bosselut et al., 2019) model, to get properties of the TENOR. For example, given a query “<Autumn,

Label	Examples
NM	瀑布注入水潭的一刹那,一朵朵白色的一浪一花腾空而起,像溅玉抛珠一般。 At the moment when the waterfall was poured into the pool, a white spray of flowers vacated, like a splash of jade beads.
NM	食堂开饭时, 全校同学像热锅上的蚂蚁一样挤成一团。 When the dining hall opened, the whole school huddled together like ants on a hot pot.
Not NM	泛着银光的大海在他身后铺展开来。 The silver-filled sea spread out behind him.

Table 5: Examples of metaphor and not metaphor in the CMC.

Category	#Books	#Tokens	#Sentences
Children	195	17M	0.58M
Chinese	336	64M	2.2M
Translated	854	121M	4.2M

Table 6: Summary of CLC.

SymbolOf>”, COMET predicts a list of properties for Autumn: “Passion, gold” etc. We then 2) construct literal expressions using the TENOR and its properties. For example, “Autumn is a symbol of passion” is obtained. 3) The literal expression is fed to SCOPE model and a simile is produced. For example, “Autumn is like a lover” is produced by SCOPE model. 4) At last, the simile are concatenate with its literal expression to form a complete NM with context: “Autumn is a symbol of passion, like a lover”.

C Meta Metric

The CMC corpus is split into training set (80%) and test set (20%) for training the classifier. We simply add a linear layer plus a binary softmax layer on the RoBERTa model as the NM classifier. The accuracy of the classifier tested on test set of CMC is 97.89%.

D More Examples

Table 7 shows generations produced by our method given different TENORS.

Text (Chinese)	Text (Translated)
爱像一缕金光，即使在黑夜也能照亮你的心灵。	Love is like a ray of golden light, which can illuminate your heart even at night.
爱像一盏明亮的夜灯，让迷途的航船找到港湾；	Love is like a bright night light, let the lost ship find the harbor.
时间像利剑一样无情的锋刃，一旦出鞘，瞬间就割断你人生的纽带。	Time is a ruthless blade like a sharp sword. Once it comes out of the scabbard, it will cut off the bond of your life in an instant.
秋天像个美人的画笔调侃着大地：世界上再没有比这更美的了。	Autumn teases the earth like a beautiful brush: there is nothing more beautiful in the world.
爱心像一片照射在冬日的光，使饥寒交迫的人感到人间的温暖。	Love is like a piece of sunshine in winter, which makes hungry and cold people feel the warmth of the world

Table 7: More generation examples of MetaGen.

Look and Answer the Question: On the Role of Vision in Embodied Question Answering

Nikolai Ilinykh and Yasmeen Emampoor and Simon Dobnik

Centre for Linguistic Theory and Studies in Probability (CLASP),
Department of Philosophy, Linguistics and Theory of Science (FLoV),
University of Gothenburg, Sweden

{nikolai.ilinykh, simon.dobnik}@gu.se, gusemampya@student.gu.se

Abstract

We focus on the Embodied Question Answering (EQA) task, the dataset and the models (Das et al., 2018). In particular, we examine the effects of vision perturbation at different levels by providing the model with either incongruent, black or random noise images. We observe that the model is still able to learn from general visual patterns, suggesting that they capture some common sense reasoning about the visual world. We argue that a better set of data and models are required to achieve better performance in predicting (generating) correct answers. The code is available here: <https://github.com/GU-CLASP/embodied-qa>.

1 Introduction

When language generation models are employed in real-world scenarios, they need to correctly perceive the environment, understand physics between objects and reason about the events in order to produce logical and correct descriptions (Lake et al., 2017). In order to study and ultimately construct such models, several language-and-vision tasks were developed including Visual Question Answering (VQA) (Antol et al., 2015; Gordon et al., 2018) and Visual Dialogue (Das et al., 2017). The advantage of such models is their ability to process visual information *jointly* with language. However, several papers following have found that **vision is often dismissed** by the model and language is much more attended to. Attempts were made to influence this bias on *the dataset side* and make the contributions of both modalities more equal. For example, Goyal et al. (2017) show that coupling questions in a VQA dataset with complementary images, which lead to different responses, makes the model learn more from vision and less from language biases. A different way of tackling the language bias in VQA datasets is to augment them with a larger variety of different question types,

generated with either a template-based method or neural networks (Kafle et al., 2017). Caglayan et al. (2019) note that there exists a dataset structure bias realised through short and repetitive texts, which in principle could inhibit gains from vision. On the other hand, many papers have proposed *models* capable of better fusion between vision and language. Zheng et al. (2020) introduce a method to learn better alignment between language and vision spaces based on reasoning over entities in texts and objects in images for the VQA task. Work on multi-modal machine translation looked at the model performance when images are replaced with incongruent scenes (Elliott, 2018) or leveraging the importance of vision modality by testing different fusion techniques (Raunak et al., 2019).

VQA models cannot be directly applied in the real world scenario due to challenges that require direct interaction of the model with the environment. Therefore the task of Embodied Question Answering (EQA) has been proposed by Das et al. (2018) which is very much different from the standard VQA. It combines question answering with a preceding navigation task in the environment, first looking for a target object that the question is about. When the agent reaches the navigation endpoint, the system answers the question based on the view from its final position. Therefore, the success of the navigation directly affects the accuracy of question answering. EQA task is much harder than VQA, because (i) the robot does not contain a human model of attention (Dobnik and Kelleher, 2016), (ii) there is no guarantee that navigation will be successful, (iii) all questions relate to home environments, which are more similar to each other than unconstrained situations in the photographs used for VQA, and (iv) questions are limited in vocabulary, scope and complexity which restricts the language and makes it even a stronger predictor. To support the latter, Thomason et al. (2019) have shown that a language-only model outper-

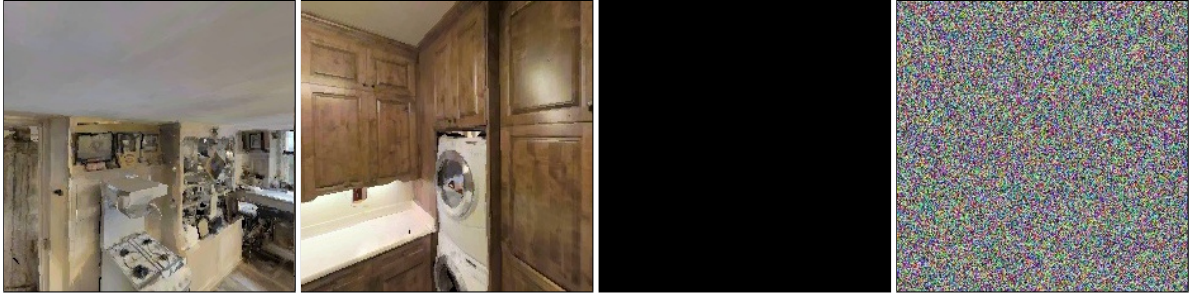


Figure 1: Example of successive removal of context, content and structure. For each removal type, we show the first frame from the set of frames that the model takes to answer the question “What color is the stove in the kitchen?”. From left to right: **original** (nothing is removed), **shuffled** (structure and content are present, but context is incorrect), **blind** (no content and context, but structure), **random** (most disturbed representation).

forms multi-modal or vision-only system during QA in the EQA task. This demonstrates a stronger need for the deeper analysis of how and to what extent vision can be even utilised in the EQA model.

While most of the existing research on EQA has focused on the navigation subtask (Wijmans et al., 2019; Yu et al., 2019; Batra et al., 2020), in this work we examine **the general role of vision for the QA in the EQA task**. In particular, we investigate how EQA model is using visual information and whether it is sensitive to visual perturbations when answering the question. First, we confirm previous results, comparing models trained and tested in different uni-/multi-modal conditions showing that just as in the VQA task, the model in the EQA task tends to hallucinate and disregard vision. Second, we turn to the examination of *how different visual disturbances affect performance of the model*. We evaluate the model with images of different types exemplified in Figure 1. The effects of various disturbances reflected in the evaluation scores will tell us how much removing context, content and (or) structure from images impacts question answering.

Our study can be viewed as *a test bed* to understand how vision is used in the EQA task. Similar benchmarks were developed for VQA (Agrawal et al., 2018) and person-centric visual grounding (Luo et al., 2022). In terms of the EQA, most of the work examined what can be used *instead* of the visual features. For example, Hu et al. (2019) show that using route structures instead of visual representations is better for the task. Schumann and Riezler (2022) found out that the model relies on properties of the environment graph much more rather than on visual features in the EQA for outdoor scenes. Different from previous studies, here we do not completely remove visual modality or compare it against other modalities. Instead, we

evaluate *the limits* of the existing EQA model when its vision is permuted. We also view the EQA task as a simple NLG task, e.g. the model is asked to map important parts in vision and language (content selection) followed by prediction of a *single* label (surface realisation). In general, the focus of this paper is to understand the interplay between different modalities used in this simple generation scenario which is also relevant for generation of longer sequences of descriptions.

2 Task Description

Models The EQA task is split into two subtasks: navigation and question answering. Below we briefly describe the models used for both subtasks, a more detailed scheme is provided in Appendix A. The navigation starts with an LSTM-based *planner* (Hochreiter and Schmidhuber, 1997) that selects an action from a pre-defined set (turn left, turn right, forward, stop) based on the question \mathbf{Q} , last action \mathbf{a}_{t-1} , last hidden state \mathbf{h}_{t-1} and visual representation $\mathbf{V}_t = F(\mathbf{I}_t)$, where F is a convolutional network (Cun et al., 1990) pre-trained on three tasks: RGB reconstruction, semantic segmentation, depth estimation. Next, the current hidden state of the planner \mathbf{h}_t , the predicted action \mathbf{a}_t and the current visual input \mathbf{V}_t are given to the *controller* that decides how many times the action has to be executed. The visual input \mathbf{V} is updated for each reiteration of the action. The controller is a simple multi-layer perceptron that returns control to the planner once it concludes that it needs a new action. The question answering module is an information fusion network. The question \mathbf{Q} is encoded by an LSTM network, while F takes N frames from the end of the navigation I_{T-N}, \dots, I_T once the agent has decided to stop (as predicted by the planner) or

the maximum number of actions $T = 100$ has been taken. Both representations are jointly attended and passed through a multi-layer classifier to predict a probability distribution across the answers.

Dataset The EQA dataset consists of automatically generated questions and answers from rules. The questions are made over visual scenes from the Matterport3D dataset (Chang et al., 2017) from which answers are generated. The authors use Habitat (Savva et al., 2019) to render the visual scenes. Each question in the dataset is replicated 15 times with different coordinates for the initial position of the agent as there is no single navigation path to the target object. There are three types of questions in the published dataset:

- colour: *What colour is the OBJ?*
- colour_room: *What colour is OBJ in the ROOM?*
- location: *What room is the OBJ located in?*

Nearly 70% of all questions are of colour_room type, $\sim 15\%$ are of colour type and the rest ($\sim 15\%$) are of location type. Placeholders *OBJ* and *ROOM* are filled with objects from dataset annotations (e.g., chair, plant) and room types (e.g., bathroom, kitchen) respectively.

Dataset and model limitations We describe several issues related to the EQA dataset. First, the quality of the rendered scenes is often poor, negatively affecting both navigation and question answering (Appendix B). Annotations of answers are sometimes questionable, including the ways the set of possible answers has been defined (e.g., limited set of possible colours in the scene) (Appendix C). A different concern is the “naturalness” of questions. Some questions are highly atypical of real interactions, e.g. why would one ask “What colour is the table in the living room?”. Another problem is that house environments are visually similar, consisting of instances of the same object classes (e.g., sofas, plants) that often share the same attributes (e.g., sofas are brown, plants are green). This also leads to an unbalanced distribution of answers: some answers (“black” and “brown”) are over-represented in the dataset, possibly allowing the model to exploit these priors, e.g. sofas are often brown. Although this dataset bias amplifies the model’s ability to answer many questions about similar objects, artificially inflating accuracy on this dataset, the same biases prevent it from correctly answering questions about objects with specific properties, which require fine-grained visual

understanding. Therefore in order to truly use vision to answer questions (e.g., when sofa is red, not brown), the model must have a *deeper* understanding of *fine-grained* visual representations, but as shown by Anand et al. (2018), the EQA models often struggle to utilise visual input. In the following sections, we will examine the level of visual understanding of the EQA model and overview problems on the dataset and modelling side that make it learn so little from vision.

3 Is language really stronger in EQA?

In the first set of experiments, we change the model’s vision stream or visual input representations. **Vis-L** is the standard EQA model (Das et al., 2018) without any perturbations on the vision side. Given the question \mathbf{Q} and N image frames, the model predicts the most probable answer a^* :

$$a^* = \operatorname{argmax}_{a \in \mathcal{A}} P(a | \mathbf{Q}, \mathbf{I}_{T-N}, \dots, \mathbf{I}_T). \quad (1)$$

For the **Blind-L** model, we keep the vision stream in the model, but change visual representations. In particular, we replace them with arrays of zeros before they are passed to the CNN for pre-processing:

$$\mathbf{I}_t = \begin{bmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{bmatrix}, \quad \mathbf{I}_t \in \mathbb{R}^{3 \times 256 \times 256}. \quad (2)$$

Finally, in the **Ø-L** model, we completely remove the vision stream and train it on questions only:

$$a^* = \operatorname{argmax}_{a \in \mathcal{A}} P(a | \mathbf{Q}). \quad (3)$$

We run all three models for 50 epochs using the official implementation¹ and choose the checkpoints with the lowest validation loss. For evaluation, we calculate accuracy (the top answer) and the mean rank (position of the correct answer in the ranked list of answers by the predicted probability distribution). We also compute Cohen’s Kappa (Artstein and Poesio, 2005) which measures the agreement between the classifier and the ground truth dataset corrected by agreement by chance which is based on the distribution of labels. A kappa close to 0

¹https://github.com/facebookresearch/habitat-lab/tree/main/habitat_baselines/il

(which ranges from 0 to 1 for agreement and 0 to -1 for disagreement) indicates that most agreement can be predicted only by knowing a distribution of labels. The higher the kappa the more the classifier is utilising additional knowledge that it has learned beyond a distribution of labels.

Metric	Vis-L	Blind-L	Ø-L
↓ Overall Mean Rank (MR)	4.352	4.454	3.685
MR, Color Room Questions	3.611	3.157	3.247
MR, Color Questions	2.693	2.261	2.304
MR, Location Questions	10.137	13.667	7.611
↑ Overall Accuracy (A)	0.38	0.323	0.362
A, Color Room Questions	0.374	0.348	0.337
A, Color Questions	0.528	0.478	0.522
A, Location Questions	0.222	0	0.278
Kappa Score	-0.005	0.014	0.024

Table 1: Results for the models both *trained* and *evaluated* with the specified settings described in Section 3. We also report results per question type. The best scores are coloured in **blue**.

The results are shown in Table 1. The **Vis-L** model has the highest overall accuracy. However, the kappa score close to 0 shows that the model has a similar performance to a model that has memorised the distribution of labels. The lower mean ranks for **Blind-L** and **Ø-L** show that they are better at approximating the correct answer than the **Vis-L** model. These models strongly learn from language since the lack of vision does not prevent them from learning from biases in the dataset, leading to higher ranks. The **Vis-L** model however needs to process vision, but it is not capable of doing that (Thomason et al., 2019). Thus vision interferes and obstructs it from learning from language biases, confusing the problematic model and leading to lower ranks of the correct answers. When breaking the results based on question types, colour question are generally the easiest to answer, followed by the colour_room and location questions. The location questions are the hardest to predict in terms of accuracy and ranking overall. Furthermore, they are also most affected by different model configurations. In particular, the results suggest that the location questions are better predicted from language alone (**Ø-L**). The **Blind-L** model has the worst ranks and the worst accuracy overall. Its inconsistent performance across question types is hard to explain. Possibly, irrelevant visual information (black images) makes it more unpredictable than no vision at all or complete vision. Although the **Blind-L** is not the optimal model, it is still not far off from the other two models due to the second

source of information - language.

Overall, we partially replicate the results of Thomason et al. (2019) and observe that vision is not that crucial. The role of language is much stronger than the role of vision, as demonstrated by the performance of the **Ø-L** model that predicts answers from questions alone. However, Frank et al. (2021) show that diminishing the importance of vision is detrimental for language tasks. Therefore in the second experiment we investigate *how different visual perturbations are utilised by the model and what are the model’s limits in learning from vision*. We are particularly interested in examining if the model is able to understand complex high-level patterns from images or does it only learn lower-level information, which is present in some form in different visual permutations.

4 “How much” vision is required?

To understand the limits of the model when utilising vision, we ask the following question: how much information can the model extract from different visual representations? We train the model according to Eqn. 1, but *evaluate* it on the vision with various levels of perturbations. In the **Eval-Shuffled** set-up, the model is provided with incorrect images for a specific question. In this case, the model gets structurally plausible representations which do not contain object(s) that the question asks about since the images depict a different house or room. We give more details about shuffling in the Appendix D. The **Eval-Blind** model has been evaluated on images which were transformed into arrays of zeros, following Eqn. 2. In **Eval-Random**, the model has been given arrays of random noise as its visual input. The image vectors were replaced by an array of the specified shape ($3 \times 256 \times 256$) that was populated with random samples from a uniform distribution:

$$\mathbf{I}_t = \begin{bmatrix} \mathbf{v} & \cdots & \mathbf{v} \\ \vdots & \ddots & \vdots \\ \mathbf{v} & \cdots & \mathbf{v} \end{bmatrix}, \quad \mathbf{v} \in [0, \dots, 1]. \quad (4)$$

Results in Table 2 demonstrate that each of the **Eval-** configurations results in lower performance compared to the baseline (**Vis-L**). However, the model performs better on both incongruent (**Eval-Shuffled**) and black (**Eval-Blind**) images rather than random noise (**Eval-Random**). This suggests that the model is using *visual patterns* to support its

Metric	Vis-L	Eval-Shuffled	Eval-Blind	Eval-Random
↓ Overall Mean Rank (MR)	4.352	5.145	5.508	6.899
MR, Color Room Questions	3.611	4.157	4.562	5.512
MR, Color Questions	2.693	3.035	3.087	3.319
MR, Location Questions	10.137	12.722	13.278	18.33
↑ Overall Accuracy (A)	0.38	0.266	0.246	0.211
A, Color Room Questions	0.374	0.264	0.258	0.258
A, Color Questions	0.528	0.307	0.217	0.194
A, Location Questions	0.222	0.222	0.222	0
Kappa Score	-0.005	0.013	0.004	-0.005

Table 2: Results for the models trained with original data (as **Vis-L**), but *evaluated* with specified conditions, described in Sec. 4). We also report results per question type. Intensity of the blue colour indicates performance of the model for the specific metric (more intensity means better performance).

prediction in some way. The performance across question types is similar to the results for models from the first set of experiments in Table 1: location questions are the hardest, colour questions are the easiest. Both experiments suggest that the visual information is not used as much as one would hope - disturbing vision or completely removing it has little effect on the overall performance, suggesting that the model exploits language more. In terms of accuracy, location questions (which have the lowest accuracy on the baseline) are affected the least by different visual input. One reason could be that the baseline is bad so there is not much room for decrease in performance. Another reason could be that there are only 15 *distinct* location question-answer pairs in the evaluation set, seven of which are also found in the training. This may be the reason for a more exploitable language bias for location questions compared to other types.

5 EQA: biases and limitations

Recently, Hirota et al. (2022) have discovered social and gender biases in the VQA dataset. In the EQA, on the other hand, the model acts in the house environments with household objects without any humans, meaning that there are no biases towards any social group. The nature of dataset problems in the EQA task is different from VQA. One of the primary problems of the EQA is the lack of the perfect navigation module that would select correct images as input to the QA module. In addition, even if navigation is perfect, there is a chance for an image to be badly rendered (Appendix B). These problems combined make the task harder and bridge it with the likes of captioning of images taken by visually impaired people (Gurari et al., 2018) instead of VQA where images are fixed and taken in perfect conditions to answer the question. Another problem is of the limited scope of automatically

generated questions and distribution of answers. In our view this directly forces the model to rely on language (which is limited and predictable) and to consider only basic visual patterns.

6 Conclusion

We looked at the Embodied Question Answering task and the corresponding dataset, focusing on how much vision is exploited by the QA module. The novelty of our study is the examination of *how* and *what* does the model learn from different types of images. Our results suggest that even if vision is not properly used, the model can extract general patterns from different visual permutations that are helpful to some degree. This means that the model could be looking at incongruent images or images with homogeneous structure (black) and answer questions correctly. Overall, we show that the model captures low-level knowledge of vision but is not capable of identifying and reasoning about specific high-level visual contexts that require understanding of scenes at a fine-grained level. Future work can improve model’s vision by implementing cognitive attention (Dobnik and Kelleher, 2016; Kruijff-Korbayová et al., 2015) or splitting the QA task into more subtasks because QA involves several inference steps and is not a simple pattern matching procedure. Using pre-trained multi-modal transformers such as LXMERT (Tan and Bansal, 2019) could also tell us whether these models are able to overcome problems related to dataset construction and image selection for the QA task in the EQA. If a performance of such a model improves then it must be the case that transformers capture some common sense knowledge through pre-training, but this could also be a hallucination of a different kind: it is hallucination because it is general V&L knowledge not the specific one arising from a particular image and text.

Acknowledgements

The research reported in this paper was supported by a grant from the Swedish Research Council (VR project 2014-39) for the establishment of the Centre for Linguistic Theory and Studies in Probability (CLASP) at the University of Gothenburg.

References

- Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. [Don't just assume; look and answer: Overcoming priors for visual question answering](#). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4971–4980.
- Ankesh Anand, Eugene Belilovsky, Kyle Kastner, Hugo Larochelle, and Aaron C. Courville. 2018. [Blindfold baselines for embodied QA](#). *CoRR*, abs/1811.05013.
- Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. [Vqa: Visual question answering](#). In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2425–2433.
- Ron Artstein and Massimo Poesio. 2005. $\kappa^3 = \alpha$ (or β). Technical report, University of Essex Department of Computer Science. Available at: <http://ron.artstein.org/publications/kappa3.pdf>.
- Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. 2020. [Objectnav revisited: On evaluation of embodied agents navigating to objects](#). *CoRR*, abs/2006.13171.
- Ozan Caglayan, Pranava Madhyastha, Lucia Specia, and Loïc Barrault. 2019. [Probing the need for visual context in multimodal machine translation](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4159–4170, Minneapolis, Minnesota. Association for Computational Linguistics.
- Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. [Matterport3d: Learning from rgb-d data in indoor environments](#). Cite arxiv:1709.06158.
- Y. Le Cun, B. Boser, J. S. Denker, R. E. Howard, W. Hubbard, L. D. Jackel, and D. Henderson. 1990. *Handwritten Digit Recognition with a Back-Propagation Network*, page 396–404. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. 2018. [Embodied question answering](#). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–10.
- Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, Jose M. F. Moura, Devi Parikh, and Dhruv Batra. 2017. [Visual dialog](#). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 326–335.
- Simon Dobnik and John D. Kelleher. 2016. [A model for attention-driven judgements in type theory with records](#). In *Proceedings of the 20th Workshop on the Semantics and Pragmatics of Dialogue - Full Papers*, New Brunswick, NJ. SEMDIAL.
- Desmond Elliott. 2018. [Adversarial evaluation of multimodal machine translation](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2974–2978, Brussels, Belgium. Association for Computational Linguistics.
- Stella Frank, Emanuele Bugliarello, and Desmond Elliott. 2021. [Vision-and-language or vision-for-language? on cross-modal influence in multimodal transformers](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9847–9857, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. 2018. [Iqa: Visual question answering in interactive environments](#). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4089–4098.
- Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. [Making the v in vqa matter: Elevating the role of image understanding in visual question answering](#). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6904–6913.
- Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. 2018. [Vizwiz grand challenge: Answering visual questions from blind people](#). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3608–3617.
- Yusuke Hirota, Yuta Nakashima, and Noa Garcia. 2022. [Gender and racial bias in visual question answering datasets](#). In *FAccT '22: 2022 ACM Conference on Fairness, Accountability, and Transparency, Seoul, Republic of Korea, June 21 - 24, 2022*, pages 1280–1292. ACM.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

- Ronghang Hu, Daniel Fried, Anna Rohrbach, Dan Klein, Trevor Darrell, and Kate Saenko. 2019. [Are you looking? grounding to multiple modalities in vision-and-language navigation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6551–6557, Florence, Italy. Association for Computational Linguistics.
- Kushal Kafle, Mohammed Yousefhusien, and Christopher Kanan. 2017. [Data augmentation for visual question answering](#). In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 198–202, Santiago de Compostela, Spain. Association for Computational Linguistics.
- Kenneth L. Kelly. 1965. [Twenty-two colors of maximum contrast](#). *Color Engineering*, 3:26–27.
- Ivana Kruijff-Korbayová, Francis Colas, Koen Hindriks, Mark Neerinx, Petter Ögren, Mario Gianni, Tomáš Svoboda, and Rainer Worst. 2015. [TRADR Project: Long-Term Human-Robot Teaming for Robot Assisted Disaster Response](#). *KI - Künstliche Intelligenz*, 29(2):193–201.
- Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. 2017. [Building machines that learn and think like people](#). *Behavioral and Brain Sciences*, 40:e253.
- Yiran Luo, Pratyay Banerjee, Tejas Gokhale, Yezhou Yang, and Chitta Baral. 2022. [To find waldo you need contextual cues: Debiasing who’s waldo](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 355–361, Dublin, Ireland. Association for Computational Linguistics.
- Vikas Raunak, Sang Keun Choe, Quanyang Lu, Yi Xu, and Florian Metze. 2019. [On leveraging the visual modality for neural machine translation](#). In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 147–151, Tokyo, Japan. Association for Computational Linguistics.
- Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. 2019. [Habitat: A platform for embodied ai research](#). In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9339–9347.
- Raphael Schumann and Stefan Riezler. 2022. [Analyzing generalization of vision and language navigation to unseen outdoor areas](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7519–7532, Dublin, Ireland. Association for Computational Linguistics.
- Hao Tan and Mohit Bansal. 2019. [LXMERT: Learning cross-modality encoder representations from transformers](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111, Hong Kong, China. Association for Computational Linguistics.
- Jesse Thomason, Daniel Gordon, and Yonatan Bisk. 2019. [Shifting the baseline: Single modality performance on visual navigation & QA](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1977–1983, Minneapolis, Minnesota. Association for Computational Linguistics.
- Erik Wijmans, Samyak Datta, Oleksandr Maksymets, Abhishek Das, Georgia Gkioxari, Stefan Lee, Irfan Essa, Devi Parikh, and Dhruv Batra. 2019. [Embodied question answering in photorealistic environments with point cloud perception](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6659–6668.
- Licheng Yu, Xinlei Chen, Georgia Gkioxari, Mohit Bansal, Tamara L. Berg, and Dhruv Batra. 2019. [Multi-target embodied question answering](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6309–6318.
- Chen Zheng, Quan Guo, and Parisa Kordjamshidi. 2020. [Cross-modality relevance for reasoning on language and vision](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7642–7651, Online. Association for Computational Linguistics.

A Baseline QA Model

Fig. 2 shows the architecture of the baseline model for question answering in the EQA task. The model consists of three parts: language encoder, vision encoder and attention across both modalities. Questions are processed by a standard LSTM network (Hochreiter and Schmidhuber, 1997) that also learns word embeddings from scratch. $B = 20$ stands for the batch size, $N = 5$ is the number of used image frames taken from the last steps of navigation, $L = 11$ is the question maximum length, and $M = 64$ is the dimension size. Note that each question representation is repeated N times. Images are represented as three-channel (RGB) 256×256 egocentric scenes from the Habitat’s image renderer. A CNN network that has been pre-trained for RGB reconstruction, semantic segmentation and depth estimation is used to process images. The fully connected layer refers to a sequence of a linear layer, a ReLU layer, and a dropout layer with $p = 0.5$. $D = 4608$ is the dimension size of the visual processing network. The output representations from the language and

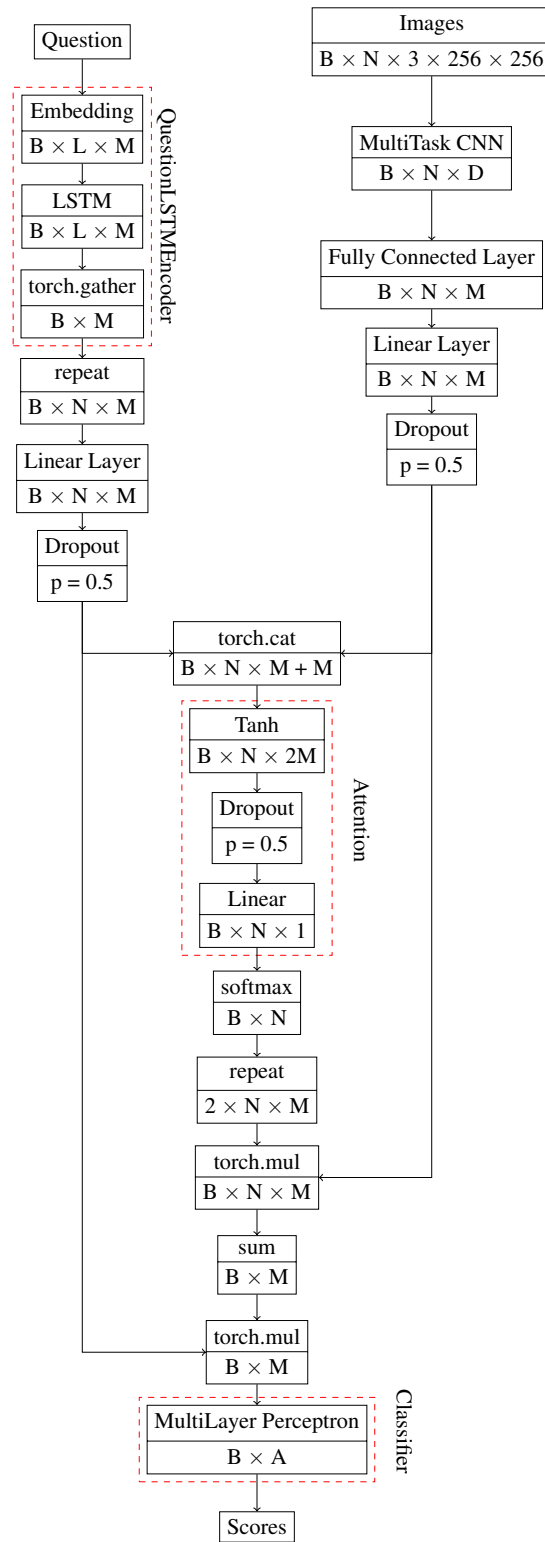


Figure 2: The baseline question answering model described in Das et al. (2018) with available implementation in Habitat-Lab, link: https://github.com/facebookresearch/habitat-lab/tree/main/habitat_baselines/il. We schematically show the key components of the model: QuestionLSTMEncoder, Attention, and Answer Classifier. The stream in the top right side corresponds to the processing of visual information.

Question: what color is the plant in the kitchen ?
Prediction: olive green
Ground truth: green



Figure 3: Example of a badly rendered scene from the EQA dataset.

Question: what color is the sofa in the living room ?
Prediction: tan
Ground truth: yellow



Figure 4: Example of a sequence of images, the question, the predicted answer and the ground-truth answer.

vision encoder are jointly attended and summed across N frames. The resulting representation is passed to a multi-layer perceptron to predict the scores across $A = 35$ possible answers. We ran all models on 4 NVIDIA GeForce GTX 1080 Ti GPUs, running time was approximately 4 hours per model. In all experiments we report results for the models with the minimal loss across 50 epochs. In our experiments we did not use any explicit tools except Habitat-Lab¹, release version 0.1.7, MIT license.

B Image Rendering Problem

While the majority of scenes are rendered properly, some of the scenes could be of poor quality. An example is shown in Fig. 3, demonstrating that the last five image frames used to answer the question include a lot of visual noise which makes the scene very confusing for a human eye. One wonders how an agent processes such poorly rendered scenes: does it rely on language information to answer the question? Note that scene annotations often include an object named “void” which is simply a black space. It is possible that the agent will encounter such confusing and uninformative space at the end of its navigation path. This could either confuse the agent or enforce better learning from the language stream. Or can the agent infer the answer from the general colours in the scene, given that the naviga-

tion often finishes at a close proximity to the target object? The example that we show is intended to demonstrate that due to the quality of the visual input, the agent might be biased to strongly learn from language and dataset biases.

C Colour Problem

The EQA dataset has been generated automatically which means that it might contain errors. An example is shown in Fig. 4, where the question answering model has answered “tan” when asked about the colour of the sofa in the living room. One could say that, when looking at the image, the sofa is indeed tan, while there is a yellow armchair next to it. It could be that the model is actually correct in its prediction for a good reason and annotations are incorrect. The problem with colour annotations is also related to the set of colours used by annotators, that is coming from Kenneth L. Kelly’s “Twenty-two colours of maximum contrast” (Kelly, 1965) with two additional colours: “off-white” and “slate grey”. This set of colours has been designed to describe situations when contrast is needed (e.g., colour coding of graphs), not necessarily to depict colours in real world with natural descriptions. For example, the set introduces “buff” and “yellowish pink”, the former one is replaced with “tan” in the EQA dataset and the latter one is simply replaced with “yellow pink”, which makes the dataset even

less natural. In addition, many colours in this set are easily confused under different lighting conditions (“white” and “off-white”, “grey” and “slate grey”), complicating the task for the question answering model.

D Example Episode

An example episode structure from the EQA dataset. We display only a part of the shortest path coordinates and viewpoint lists. In **Eval-Shuffle**, shuffling is performed by modifying the original set of image frames and creating a new one. We show an example of one navigation episode from the EQA dataset below. A single episode includes a `question` field, which includes the question, answer, question type, and answer token IDs. We shuffle these `question` fields (line 68 in the example structure) across different episodes. Note that the authors of the dataset duplicated questions across multiple episodes, which, however, have different navigation paths to the target. This has been implemented in order to ease the navigation task since there is no single correct navigation to the target object. We acknowledge that it could be possible that an episode with a shuffled question still has a valid set of last N image frames, but this possibility is low – for a single question, this probability is less than one percent.

```

1 {'episode_id': '640',
2   'scene_id': 'mp3d/5LpN3gDmAk7/5LpN3gDmAk7.glb',
3   'start_position': [15.50573335967819,
4     ↪ -0.7660300302505512, 8.392731789742543],
5   'start_rotation': [-5.312086480921031e-17,
6     ↪ -0.8526401643962381,
7     ↪ 0.522498564647173],
8   'info': {'bboxes': [{'type': 'object',
9     'box': {'centroid': [13.2358, -14.5238, 0.497693],
10      'a0': [1.0, 0.0, 0.0],
11      'a1': [0.0, 1.0, 0.0],
12      'a2': [0.0, 0.0, 1.0],
13      'radii': [0.593273, 0.243441, 1.68627],
14      'obj_id': 305,
15      'level': 0,
16      'room_id': 18},
17     'name': 'door',
18     'target': True},
19     {'type': 'room',
20      'box': {'centroid': [10.874245, -11.97072, 0
21        ↪ .5380600000000001],
22      'a0': [1.0, 0.0, 0.0],
23      'a1': [0.0, 1.0, 0.0],
24      'a2': [0.0, 0.0, 1.0],
25      'radii': [3.1686549999999998, 3.26178, 1.95437],
26      'room_id': 18,
27      'level': 0},
28      'name': ['kitchen'],
29      'target': False}],
30   'question_meta': [{'name': 'colour', 'diffuse':
31     ↪ 'grey'}],
32   'question_answers_entropy': 0.8303560860446519,
33   'level': 0},
34   'goals': [{'position': [13.2358, 0.4976929999999973, 14
35     ↪ .5238],
36     'radius': 0.6412771421234348,
37     'object_id': 305,
38     'object_name': 'door',
39     'object_category': 'object',
40     'room_id': 18,
41     'room_name': 'kitchen',
42     'view_points': [{'position': [12.985883260576134,
43       ↪ -1.246680130110505,

```

```

41     ↪ 14.494095338174798],
42     'rotation': [-2.855981544936522e-28,
43     ↪ -0.7071067811874078,
44     ↪ -0.0,
45     ↪ 0.7071067811856873]],
46     ...
47     {'position': [13.089462756345679,
48       ↪ -1.246680130110505, 13.976197859327065],
49     'rotation': [-1.2227381688226952e-16,
50     ↪ -0.8910065241891411,
51     ↪ 0.45399049973802935]]}],
52   'start_room': 'R22',
53   'shortest_paths': [[{'position': [15.50573335967819,
54     ↪ -0.7660300302505512,
55     ↪ 8.392731789742543],
56     'rotation': [-5.312086480921031e-17,
57     ↪ -0.8526401643962381,
58     ↪ -0.0,
59     ↪ 0.522498564647173],
60     'action': 2),
61     ...
62     {'position': [13.042462387438766,
63       ↪ -0.7660300302505512, 13.951177365325918],
64     'rotation': [-1.2227381690007914e-16,
65     ↪ -0.891006524228339,
66     ↪ -0.0,
67     ↪ 0.45399049967190386],
68     'action': 3}],
69   'question': {'question_text': 'what colour is the door
70     ↪ in the kitchen?',
71     'answer_text': 'grey',
72     'question_tokens': [4, 5, 6, 7, 19, 9, 7, 10],
73     'answer_token': [0, 0, 0, 0],
74     'question_type': 'colour_room'}}

```


Strategies for Framing Argumentative Conclusion Generation

Philipp Heinisch

Bielefeld University

pheinisch@techfak.uni-bielefeld.de

Anette Frank

Heidelberg University

frank@cl.uni-heidelberg.de

Juri Opitz

Heidelberg University

opitz@cl.uni-heidelberg.de

Philipp Cimiano

Bielefeld University

cimiano@techfak.uni-bielefeld.de

Abstract

Generating an argumentative conclusion from a set of textual premises is a challenging task, due to a large range of possible conclusions. In order to provide a conclusion generation model with guidance towards generating conclusions from a certain perspective, we explore the impact of conditioning the model on information about the desired framing. We experiment with conditioning generation via generic frame classes as well as with so-called issue-specific frames. Beyond conditioning the model on a desired frame, we investigate the impact of strategies to further improve the generated conclusion by i) an informative label smoothing method that dynamically smooths one-hot-encoded reference conclusion vectors as a regularization mechanism, and ii) a conclusion reranking strategy based on referenceless scores at inference time. We evaluate the benefits of our methods using metrics for automatic evaluation complemented with an extensive manual study. Our results show that frame-guided conclusion generation is beneficial: it increases the ratio of valid and novel conclusions by 23%-points compared to a baseline without frame information. Our work indicates that i) by injecting frame information, conclusion generation can be directed towards desired aspects and ii) at the same time it can be manually confirmed to yield more valid and novel conclusions.

1 Introduction

Argument mining enables systems to automatically retrieve (Wachsmuth et al., 2017a), analyse (Becker et al., 2020), classify (Trautmann et al., 2020), rank (Wachsmuth et al., 2017b) or summarize (Bar-Haim et al., 2020) arguments on a controversial topic. In line with growing amounts of user-generated argumentative content, this field is intensely researched and bears the potential to support humans in deliberation (Fromm et al., 2019).

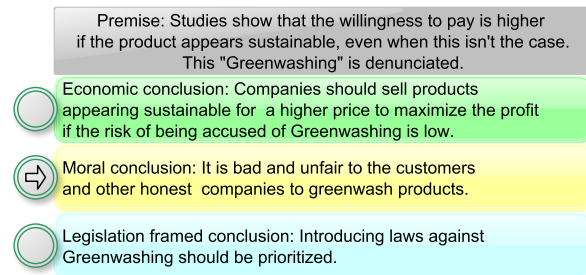


Figure 1: Example argument. All three conclusions are appropriate, but are framed in different ways.

An argument can be conceptualized as a pair of premise(s) and a conclusion. At the core of an argument lies the inferential link between the premises (evidences) and the conclusion. Current systems capture this inferential link only to a limited extent, e.g. by predicting whether a premise supports or attacks a given claim (Cocarascu et al., 2020). While approaches such as Paul et al. (2020) establish chains of background knowledge that characterize the link between premises and conclusions, such methods are limited to analyzing *existing* arguments. To better understand whether computational systems are able to draw inferences from premises towards conclusions, in this work we study the problem of automatic conclusion generation. Being able to automatically *generate* conclusions bears great potential: not only could we retrieve arguments and make their unstated conclusions explicit – such a method would also allow us to generate novel arguments in a debate, thereby supporting deliberation – by raising novel conclusions from different perspectives.

Yet, the process that infers a conclusion from a set of premises is underspecified, since different conclusions may be drawn from a set of premises, depending on different viewpoints. An example is illustrated in Figure 1. It shows the importance of being able to reflect on a topic under discussion from various perspectives (de Vreese, 2005).

Two approaches have been used to describe the different perspectives or "framings" of a discussion. Several authors (Neuman et al., 1992; Boydston et al., 2014) have proposed to work with a fixed set of manually defined frames, so-called *generic frames*. Others, adopting a more open approach (de Vreese, 2005), have proposed to rely on a vocabulary of *issue-specific frames* that vary from debate to debate, are more fine-grained, and can be provided by users to cluster their arguments in a certain debate (Ajjour et al., 2019). Building on these two notions of *framing*, we investigate which type of frame information is most effective to guide a conclusion generation model.

Previous work has attempted to reconstruct a missing conclusion by identifying the "main target" in the premises (Alshomary et al., 2020). Other work has made use of pretrained sequence-to-sequence transformer language models fine-tuned on argumentative datasets (Syed et al., 2021; Opitz et al., 2021; Gurcke et al., 2021). However, the question of how to tailor a generated conclusion to a particular frame has not been systematically explored, a gap that we address with this paper. Our contributions are the following¹:

- i) We present a framework and method based on autoregressive transformer-based decoding to study how the generation of (textual) conclusions can be controlled by integrating information about the desired frame as input. We explore different frame granularities separately and in combination: generic frames as defined by Boydston et al. (2014) and issue-specific frame labels.
- ii) We present results on the issue-specific frames dataset by Ajjour et al. (2019), showing improvements resulting from conditioning on a desired frame, through i) automatic evaluation, as well as ii) a study relying on human annotators rating *validity*, *novelty* and *frame relatedness* of the conclusion.
- iii) We investigate additional strategies to guide the conclusion generation model towards selecting an appropriate conclusion, using a label-smoothing method applied at *training time*, and two strategies (frame-sensitive decoding and conclusion reranking) applied at *inference time*. These additional methods yield further improvements, while highlight-

ing an interesting trade-off between validity and novelty of the generated conclusions.

2 Related work

While massive amounts of user-generated arguments are available in various debate portals or writing platforms, these arguments are often incomplete, missing an explicitly stated conclusion or lacking essential premises. Such omissions are frequent and often due to rhetorical reasons (Rajendran et al., 2016; Becker et al., 2021). However, arguments lacking an explicit conclusion create challenges for downstream processing tasks (Opitz et al., 2021; Alshomary et al., 2020; Gurcke et al., 2021). Thus, prior work has investigated approaches to make conclusions explicit.

First approaches in this direction attempt to extract missing parts by copying from similar or related arguments, or by applying common, hand-crafted argument patterns (Rajendran et al., 2016; Reisert et al., 2018). Yet, these approaches are limited due to the variety of human argumentation and do not generalize well to novel topics.

More recent works leverage sequence-to-sequence transformer language models: Syed et al. (2021) is the first approach known to us that relied on transformer models to generate conclusions given premises. They relied on a pretrained BART model showing that it is able to create premise-related text. However, their manual study shows that 14-36% of the generated conclusions are valid, e.g. by rephrasing the premise, but only 4-6% are informative. Opitz et al. (2021) also show that state-of-the-art fine-tuned transformer language models processing plain premises tend to generate conclusions lacking in novelty or validity, and proposed ways to assess their novelty and validity using AMR-based similarity metrics. Finally, Gurcke et al. (2021) explored whether the sufficiency of conclusions can be assessed with BART, and find problems with insufficient reference conclusions – with ensuing challenges in generating and evaluating valid and novel conclusions.

Prior work also investigated whether the quality of generated conclusions can be improved by conditioning a language model exclusively on topic and frame. Schiller et al. (2021) show that claims generated by such a conditional transformer language model are in general of high quality.

However, none of the approaches mentioned so far has attempted to directly control the framing of

¹Our code is available on GitHub: [phhei/ConclusionGenerationWithFrame](https://github.com/phhei/ConclusionGenerationWithFrame)

a conclusion by conditioning the model via a given premise *and* the desired frame, a gap we close in this paper. We investigate different ways of encoding the frame and experimentally investigate the impact of these guides using automatic and human evaluations.

3 Datasets

To study the impact of controlling conclusion generation by conditioning on the desired frame, we rely on two datasets. One is the Media-Frames dataset, which relies on an inventory of 15 generic frames originally proposed by [Boydston et al. \(2014\)](#). The second dataset, produced by [Ajjour et al. \(2019\)](#), does not rely on a fixed set of frames, but on user-provided frames – so-called issue-specific frames. Details of both datasets are given below.

3.1 Media-Frames dataset

The Media-Frames dataset by [Card et al. \(2015\)](#) consists of 17,826 newspaper articles on three policy issues (*immigration, smoking and same-sex marriage*) annotated with the generic *Media Frames* defined by [Boydston et al. \(2014\)](#). The set of *Media Frames* contains 15 different frame classes: i) Economic, ii) Capacity and resources, iii) Morality, iv) Fairness and equality, v) Legality, constitutionality and jurisprudence, vi) Policy prescription and evaluation, vii) Crime and punishment, viii) Security and defense, ix) Health and safety, x) Quality of life, xi) Cultural identity, xii) Public opinion, xiii) Political, xiv) External regulation and reputation, as well as xv) Other. The annotation of frame information was performed in several rounds by selecting text spans and assigning them to one of the 15 frame classes, which yielded an inter-annotator-agreement between 0.29 and 0.6 according to Krippendorff’s α . To increase the reliability of the data, we rely on only those instances for which at least two annotators agree on the corresponding frame. The resulting subset contains 21,206 samples.

3.2 Argument dataset with issue-specific frames

[Ajjour et al. \(2019\)](#)’s dataset contains 12,326 arguments that were annotated with user-generated issue-specific frame labels – tags that can serve to cluster arguments in a debate to better overview the controversial aspects. The data is crawled from

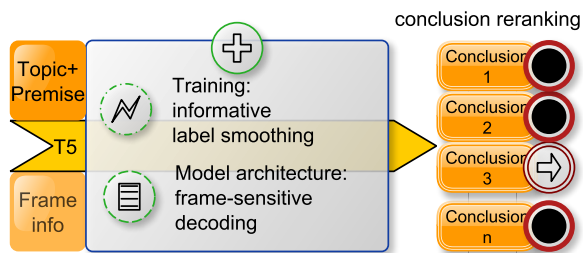


Figure 2: Overview of contributions: Frame-sensitive conclusion generation by frame-sensitive decoding, informative label smoothing and conclusion reranking.

Debatepedia² and consists of 365 different topics. In total, the label set comprises 1,623 different frames labels. Out of these, only 330 occur in two or more topics, which indicates that there is a substantial long tail of labels that occur only a few times.

4 Methods

We rely on a sequence-to-sequence encoder/decoder architecture that encodes the topic and the premise, and autoregressively decodes the conclusion. We examine whether and how the generation can be conditioned by enriching the input with information about the desired frame. We investigate i) the explicit encoding of frames as part of the input (4.1) and ii) injection of prior generic frame knowledge by adjusting the output of the language model (4.2). Moreover, we also propose more fine-grained methods: iii) an informative label smoothing training technique and iv) a conclusion reranking approach (4.3). The label smoothing approach attempts to push the model to generating a conclusion that is specific for the given desired frame, while the conclusion reranking method re-ranks potential conclusion candidates using shallow and argumentation-inspired metrics.

4.1 Explicit encoding of frames

To condition conclusion generation on a frame, we encode the frames (issue-specific and generic frames) as part of the input, as pictured in Figure 3. The input to the transformer model uses additional separators and looks as follows:

```
summarize
[T] topic [/T]
[Fis] issue-specific frame [/Fis]
[Fgm] generic frame (argument) [/Fgm]
[Fgi] generic frame (conclusion) [/Fgi]: premise.
```

²<http://www.debatepedia.org>

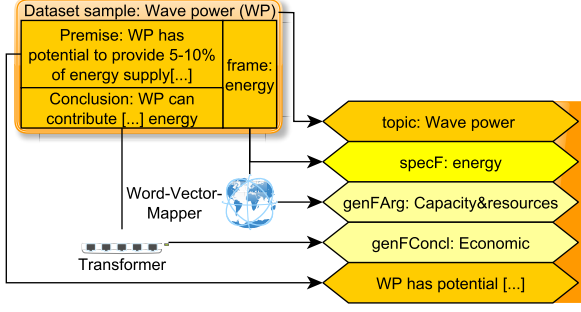


Figure 3: Different input parts for the conclusion-generating language model showing an abbreviated sample of the dataset of Ajjour et al. (2019)

Hereby, *topic* is the debate title as contained in the dataset of Ajjour et al. (2019). The *issue-specific frame* (*specF*) is the frame label as described by users for each argument in this dataset. For determining the *generic frame* (*argument*) (*genFArg*), we map each generic frame class label and the issue-specific frame into a low-dimensional semantic vector space by semantically aggregating the word-vectors as proposed by Heinisch and Cimiano (2021). We select the generic frame label that is closest in this vector space to the given issue-specific frame. Finally, we propose a second approach to inferring a generic frame denoted by *generic frame* (*conclusion*) (*genFConcl*) by using a transformer model trained on the Media-Frames dataset (Appendix A.1) that predicts the corresponding frame for the conclusion.

4.2 Frame-sensitive decoding

Our goal is to increase the likelihood that a generated conclusion is frame-specific. For this, we increase the probability that, at decoding time, the model outputs tokens that are associated with the given frame. To achieve this, we follow a finding of Naderi and Hirst (2017) who measured a correlation between particular uni- and bigrams and certain generic frames in the Media-Frames dataset. For example the \$-sign often occurs in an economically framed text. We can use this frequencies to inject frame-specific prior knowledge by adjusting the output logits o of the transformer. With this modification we can directly influence the sequence-to-sequence decoding, as shown in equation (1),

$$o'_v = \frac{o_v}{2} + o_v \left(h(o)_v \frac{\log(tf_{D_f}(v) + 1)}{\max_{\tilde{v} \in V} \log(tf_{D_f}(\tilde{v}) + 1)} \right) \quad (1)$$

$$\forall v \in V, o \in \mathbb{R}^{|V|}, h \mapsto [0, 1]^{|V|}$$

where v is a vocabulary element, $h(o)$ a parametrizable function that maps the logit values to a range of $[0, 1]$, and $tf_{D_f}(v)$ the term frequency of v in documents framed with the generic frame f . Specifically, we set the new output logit o'_v for v to half of the model's logit output, and add this logit's value scaled by its normalized frame-frequency in combination with the overall predicted logits. In this way, a higher frequency of a given word v in frame f results in a higher added value. As a result, we expect the model to prefer generating tokens that are likely to occur in the desired frame f , while dispreferring tokens that are unlikely to occur in texts framed with f .

4.3 Additional strategies to boost frame-sensitive conclusion generation

As a further option to conditioning the autoregressive generator to an explicitly encoded frame in the input and including frame-relevant word knowledge, we now analyse the impact of two strategies that aim to move the generated conclusion closer to the reference conclusions. Specifically, we apply an *informative label smoothing technique* and a *conclusion reranking strategy*.

Informative label smoothing During fine-tuning the language model for conclusion generation, we apply a regularization technique proposed by Szegedy et al. (2016) that modifies the computation of the cross-entropy loss by smoothing each one-hot-encoded conclusion token vector \vec{y} , transforming it into an token vector \vec{y}' that distributes part of the probability mass to the whole vocabulary. Given a smoothing strength parameter $\lambda \in [0, 1]$ and the token sequence of the reference conclusion $c = \{w_1, \dots, w_n\}$, the one-hot-encoded vector y_{w_i} for each token w_i is transformed as follows:

$$y_{w_i}' = \left(\frac{\lambda}{V}, \dots, 1 - \lambda + \frac{\lambda}{V}, \dots, \frac{\lambda}{V} \right) \quad (2)$$

While λ is fixed for all tokens w_i in the approach of Szegedy et al. (2016), we propose an ex-

tension that uses on a token-specific λ' that multiplies λ by two token-specific factors. The first factor (controllable by $\delta \in [0, 1]$) scales λ proportionally to the term frequency $tf(w_i)$. Thus, the more frequent the token w_i is, the higher λ' and thus the more is the output reference vector spread and further away from a one-hot encoded vector. The second factor (controllable by $\psi \in [0, 1]$) scales λ inverse proportionally to the frame-specific term frequency $tf_{D_f}(w)$ with which the token occurs in frame f . Thus, the more frequent the token w_i in the frame f , the lower the value of λ' and thus the more is the distribution centered on token w_i ³. Overall, we adjust the smoothing strength for each w_i as follows:

$$\begin{aligned} \lambda'(w_i) = & \lambda \\ & * \left(1 - \delta + \left(2\delta \frac{\log(tf(w_i)+1)}{\max_{v \in V} \log(tf(v)+1)} \right) \right) \\ & * \left(1 - \psi + \left(2\psi \left(1 - \frac{tf_{D_f}(w_i)}{\max_{v \in V} tf_{D_f}(v)} \right) \right) \right) \end{aligned} \quad (3)$$

Conclusion reranking: selecting the most appropriate conclusion We explore a conclusion reranking strategy inspired by Hua and Wang (2020), to choose a proper conclusion among different beam search traces. Given a set of conclusion candidates $\mathcal{C} = \{c_1, c_2, \dots, c_n\}$ and a set of automatically calculated reference-less scores $\mathcal{S}(c) = \{s_1(c), \dots, s_m(c)\}$ for each candidate c , we select the conclusion c' that maximizes a weighted linear combination of the reference-less scores as indicated in the following equation:

$$\begin{aligned} c' = \arg \max_{c \in \mathcal{C}} \sum_{i=1}^m \omega_i s_i(c) \\ \text{with } \omega = (\omega_1, \dots, \omega_m) \\ \text{as a fixed pretrained scalar vector.} \end{aligned} \quad (4)$$

We formalize the problem of optimizing the ω -vector as a linear optimization problem⁴ in an additional Ω -optimization split, determining the ω -vector by the following equation that minimizes the gap between the reference-less metric scores \mathcal{S} and r metric scores $\tilde{\mathcal{S}}$ using the reference:

$$q_{min}(\Omega) = \sum_{c \in \Omega} \left| \sum_{i=1}^m \omega_i s_i(c_i) - \sum_{h=1}^r \frac{\tilde{s}_h(c_h)}{r} \right| \quad (5)$$

³To avoid too small text corpora of a particular frame, the second factor requires generic frame information in the input, otherwise the second factor is disabled

⁴We also tried more complex learning models, for example SVMs. However, more complex models did not outperform the linear regression on average, while increasingly lacking in interpretability.

Examples of metrics which do not consider the reference are listed in A.2.

5 Experiments and Evaluation

In our experiments we investigate the impact of conditioning the generation of an argumentative conclusion, given a topic and premise, on different frame labels provided as part of the input, in addition to our semantic- and frame-sensitive model adjustments. We explore the influence of different types of frame information – the original issue-specific frame labels and the derived generic frame classes – combined with our model variants. We perform automatic evaluation of the generated conclusions, relying on similarity to the reference conclusion as evaluation measure. We also carry out a comprehensive manual evaluation in which annotators rated the validity and novelty of generated conclusions, as well as their closeness to the desired frame.

5.1 Experimental Setup

We follow previous work by Opitz et al. (2021) and rely on T5 (Raffel et al., 2020) (large version) as transformer language model, as implemented in the huggingface library (Wolf et al., 2020).

We use the argument dataset by Ajjour et al. (2019) (Section 3.2), which we divide into splits of 80%, 10%, 5% and 5% of the samples for training, development, Ω -optimization split for conclusion reranking and test, respectively, without overlapping topics.

We test different frame configurations, including subsets of our three frame specifications. The abbreviation *specF+genFArg*, e.g., symbolizes a fine-tuned model that receives the issue-specific and generic frame (argument) as frame information for each sample.

Training: We train between 2 and 12 epochs, where we stop the training process after the validation loss increases in two consecutive epochs. We use an initial learning rate of 2e-4 and decrease it during the training in a step-wise fashion with a factor of 0.975 every 32 steps including a minor weight decay of 1e-7. For our frame-sensitive decoding strategy that uses the Media-Frames dataset as prior knowledge base, we set the function h in Equation 1 to the softmax function. For our informative label smoothing we experimented with different parameters and got strong results with a general label smoothing factor of

$\lambda = 0.1$, a dynamic smoothing factor of $\delta = 0.4$, and a generic frame smoothing factor of $\psi = 0.5$ (Equation 3). The frame-sensitive decoding and the frame-sensitive component of the informative label-smoothing considers the generic frame (preferred from argument) part in the input. If no generic frame information is given in the input, these two frame-sensitive adjustments are deactivated.

Inference: We apply nucleus sampling as proposed by Holtzman et al. (2020) (considering tokens covering 92.5% probability, but max. 50 different tokens, by five beams or twelve in case of conclusion reranking). The temperature is set to 0.75 or 1.1 in the case of conclusion reranking to increase the word diversity. For conclusion reranking, we developed and considered a variety of reference-less scores. We consider shallow surface cues of the conclusion candidate, such as the length (also in ratio to the premise length), the ratio of stop words, the existence of conclusive trigger words such as "should" for normative conclusions, and also non-shallow metrics. To measure the grammaticality of the generated conclusion, we use the GRUEN-score (Zhu and Bhat, 2020). Furthermore, we check deep argumentative characteristics, for example the argumentative relation between the premise and the conclusion, the BERTscore between premise and conclusion candidate, to avoid copies or completely unrelated conclusions, as well as whether the generated conclusion candidate matches the desired frame, if available. We list and further describe all used reference-less metrics in the Appendix A.2. We test two different variants of conclusion reranking. The first optimizes the aggregation of the ROUGE-1, ROUGE-L, as well BERTscore, and thus the similarity to the reference conclusion on the Ω -optimization split, called *frame-insensitive conclusion reranking*. The second *frame-sensitive* variant in addition optimizes the automatic frame-relatedness scores of the selected conclusions on the Ω -optimization split.

Evaluation: We rely on a mixture of automated scores, such as the token-based ROUGE-score and the BERTscore⁵ (Zhang et al., 2020) measuring the semantic similarity between generated and reference conclusions.

⁵rescaled f_1 , using the 18th layer of microsoft/deberta-large-mnli without an idf-weighting

Evaluating the generated conclusions with respect to their references using automatic metrics might, however, penalize valid conclusions that differ substantially from the reference. We therefore also perform manual evaluation with human annotators on 30 randomly selected arguments from the test-split⁶. The annotators were paid for their work and are not authors of the paper. Each reference conclusion, random and generated conclusion is annotated three times by the same three annotators in three consecutive rounds with respect to the following dimensions: (1) **Validity:** Is the conclusion justified based on the premise?, (2) **Novelty:** Does the conclusion contain premise-related novel content that is not part of the premise?, and (3-4) **issue-specific frame / generic frame (argument):** Is the conclusion directed towards the given frame? To avoid different scale interpretations, we allow only the answers {yes, no, can't decide}. In an additional pairwise setting, presenting two conclusions, we ask whether one (and if so, which) conclusion is better in view of the rated aspect. We hide the source of the presented conclusions (reference, random or the generating model configuration) to avoid bias. Further details on the manual study are given in Appendix A.3.

5.2 Results

Impact of conditioning conclusion generation on provided frame information Table 1 shows the results of the automatic evaluation of the generated conclusions compared to their reference conclusions, for different variants of frame information provided as part of the input. We report results for three evaluation measures: ROUGE-1, ROUGE-L, and BERTscore (F1-score). As baseline, we rely on a model version that only relies on premise and topic as input, but does not include any frame information (*no frame*). We can observe that adding information about the frame in the three specifications *specF*, *genFArg*, *genFConcl* has a positive impact on the generated conclusions, increasing results between 1.1 and 4.5 points for ROUGE-1, between 1.1 and 3.9 points for ROUGE-L, and between 1.0 and 4.4 points for BERTscore. The single frame specification with the most signifi-

⁶The frame distribution of the test set is similar to the frame distribution of the selected samples, having most "other" (40%) and "economic" (17%) generic frames (argument).

Configuration	Rouge1	RougeL	F1-BERTs.
no frame	29.1	26.4	29.4
specF	+2.1	+1.8	+2.2
genFArg	+1.6	+1.2	+1.9
genFConcl	+2.5	+1.9	+1.5
genFArg+genFConcl	+1.1	+1.1	+1.1
specF+genFArg	+1.3	+1.1	+1.0
specF+genFConcl	+1.9	+1.5	+2.0
all 3 frames	+4.5	+3.9	+4.4

Table 1: Automatic scores for various frame configurations (issue-specific frame, generic frame from argument, generic frame from conclusion) without informative label smoothing and conclusion reranking

Configuration	Val	Nov	Both	spec-f	gen-f
random	0	7	0	10	33
no frame	50	50	17	67	78
specF	67	37	10	90	89
genFArg	73	37	10	87	83
genFConcl	67	50	13	77	78
specF+genFArg	40	63	7	80	72
specF+genFConcl	60	47	20	77	78
all 3 frames	70	40	10	83	83
reference	73	73	47	83	83

Table 2: Manual evaluation study: ratio of conclusions fulfilling the criteria of Validity, Novelty, both validity and novelty, and relatedness to the target issue-specific frame and the generic frame (argument), based on the majority votes for various frame configurations (issue-specific frame, generic frame from argument, generic frame from conclusion), random and human-written reference conclusions, in %.

cant impact is the generic frame (conclusion) (*genFConcl*) for ROUGE-1 and ROUGE-L and the issue-specific frame (*specF*) for BERTscore. Considering combinations of two frame specifications (*genFArg+genFConcl*, *specF+genFArg*, *genFConcl+specF*) yields worse results in all cases, compared to using a single source of information. However, using all three frame specifications yields the best result, with improvements of 4.5, 3.9, and 4.4 points for ROUGE-1, ROUGE-L, and BERTscore, respectively.

Table 2 shows the results of the manual evaluation, where three human raters decided whether the generated conclusion is i) valid, ii) novel, iii) directed towards the issue-specific frame and iv) directed towards the generic frame (argument). The table shows the percentage of conclusions for which the majority of annotators agree that the conclusion is valid/novel/directed towards the desired frame. Manual assessment of a *random* conclusion (sampled from all generated and reference conclusions) and of the *reference* conclusion pro-

vide a lower and upper bound for our approach. The results of the manual evaluation corroborate that each *single frame* configuration has a positive impact on validity between +17% to +23% points – at the expense of no improvement or even decrease in novelty. Providing frame information as input also yields an increase in frame-relatedness (up to +23% points). For combinations of two or more frame specifications we see a mixed pattern: a decrease in validity (−10% points) and increase in novelty (+13% points) for *specF+genFArg* and the reverse pattern for *specF+genFConcl* and *specF+genFArg+genFConcl* (+10%/+20% points regarding validity and −3%/−10% points regarding novelty). However, we observe that in view of generating both valid and novel conclusions, all configurations except for *specF+genFConcl* (+18%) perform below the unframed baseline. At the same time, all configurations clearly improve upon the baseline in generating a conclusion that fits the desired issue-specific frame (see Table 2), with improvements ranging from +10% (*specF+genFConcl*) to +23% (*specF*) points. Regarding the frame relatedness to the generic frame, we see clear improvements over the baseline for 3 out of 6 configurations, ranging from +5% (*genFArg*, *specF+genFArg+genFConcl*) to +11% (*specF*) points.

Below, we investigate the impact of further strategies on the four configurations that were rated as best with respect to a single dimension: *specF* (for *specF + genFArg*), *genFArg* (for validity), *specF+genFArg* (for novelty), and *specF+genFConcl* (for both validity and novelty).

Impact of strategies to boost frame-sensitive conclusion generation To measure the impact of our strategies for boosting frame-sensitive conclusion generation, the annotators were asked to rate the validity and novelty of the conclusions in a pairwise setting with and without activated label smoothing and/or conclusion reranking, and had to rate whether they found an increase, tie or decrease of novelty and/or validity.⁷ Table 3 shows the results of this further manual evaluation, where next to *Val*, *Nov* and *Both* we show the absolute improvements in automatic BERTscore for each strategy.

⁷Our annotators evaluated up to 60 samples for conclusion reranking: the 30 arguments from the first annotation rounds + 30 new arguments for input variants: *no frame*, *specF+genFArg*, *specF+genFConcl*.

Configuration	+ inf. label smoothing				+ conclusion reranking							
					frame-insensitive				frame-sensitive			
	BERT	Val	Nov	Both	BERT	Val	Nov	Both	BERT	Val	Nov	Both
no frame	+4	+13	+23	+10	+13	+10	-13	-5	n/a	n/a	n/a	n/a
specF	+7	-13	+23	+7	+8	+27	-17	+3	+7	+9	-13	0
genFArg	+2	-33	+23	0	+8	+27	-10	0	+7	+20	-10	+3
specF+genFArg	+2	+3	-27	0	+10	+20	-20	-3	+8	+17	-7	0
specF+genFConcl	+8	+30	+13	+13	+11	+8	-17	-2	+8	+13	-12	-3

Table 3: Evaluation of additional strategies for boosting frame-sensitive conclusion generation automatically (F1-BERTscore) and manually (majority votes per conclusion in Validity, Novelty, Both) for various frame configurations (issue-specific frame, generic frame from **argument**, generic frame from **conclusion**). The deltas measure the difference to the next lower model complexity (w/o any additional strategy/ only informative label smoothing) in %.

Informative label smoothing has a positive impact w.r.t. to the BERTscore (+2% to +8%). With respect to validity, it improves results in 3 out of 5 configurations, while with respect to novelty, it improves results in 4 out of 5 configurations in the range of +13% to +23%. We thus see a clear positive impact on novelty.

Conclusion reranking improves the BERTscore in all configurations, both in the frame-insensitive (+8% to +13%) and the frame-sensitive variant (+7% to +8%). Both variants have a positive impact on validity, improving results between +8 and +27% (frame-insensitive variant) and between +9 and +20% (frame-sensitive variant). However, both variants do not consistently improve the number of conclusions that are regarded as both valid and novel across configurations, with differences ranging from -5% to +3%.

5.3 Discussion

Regarding the impact of conditioning conclusion generation by providing information about the desired frame, our results of both automatic and manual evaluations are generally positive. We see a clear improvement in the framing and the similarity of the generated conclusions to their reference conclusions. The results of our manual evaluation clearly point to a trade-off between *generating a valid vs. novel conclusion*, showing that it is very challenging to generate conclusions that fulfill both criteria (novelty and validity). For example, providing information targeting an issue-specific frame (*specF*) increases validity by 17% points while decreasing novelty by 13% points at the same time. There are other configurations, however, that resolve this trade-off better. The combination of issue-specific as well as conclusion-retrieved generic frames

(*specF+genFConcl*) yields the best results in generating a conclusion that is both valid and novel (20%), outperforming the *no frame* baseline by 3% points. This configuration leverages information from the two different types of frames, providing the model information at different and thus complementary levels of granularity as proposed by [Heinisch and Cimiano \(2021\)](#).

Overall, the best configuration in terms of validity, judging from our manual majority votes, is the version that relies on issue-specific frame information as input in combination with informative label smoothing and frame-sensitive conclusion reranking (87%). Regarding novelty, the best configuration combines the issue-specific and generic frame (argument) with informative label smoothing but without conclusion reranking (67%). The configuration that excels in generating both novel and valid conclusions is the one that enriches the input with the issue-specific frame as well as the generic frame (conclusion), again using only informative label smoothing (40%).

In general, assessment by way of BERTscores does not correlate well with manual assessment of validity and novelty. While BERTscores improve in all cases when applying informative label smoothing and especially conclusion reranking (up to 37.6), the manual evaluation of validity and novelty in those configurations is quite mixed. Many configurations improve validity at the cost of novelty and the other way round. In general, informative label smoothing has a positive impact on novelty. It seems that preferring conclusions that include tokens that frequently occur with the desired frame is driving the model to leave its comfort zone and take risks in generating conclusions with novel elements with the downside that some of these conclusions seem not to be perceived as valid. In contrast, conclusion reranking, which

Wave power – premise:	
	Wave power has the potential to provide 5-10% of US energy supply, according to the New York Times. (<i>issue-specific frame: energy / generic frame (argument): Capacity and resources</i>)
reference	Wave power can contribute a significant amount of energy
no frame	Wave power can significantly increase energy supply
+smooth	Wave power has the potential to replace coal
genFArg	Wave energy has the potential to power the US
specF+genFArg	Wave power is a major source of clean energy
+smooth	Wave power can supply a significant amount of energy
+concl.rerank	Wave power can provide 10% of US energy supply

Table 4: Case study showing the effects of different configurations, including informative label smoothing and frame-insensitive conclusion reranking

learned to optimize primary the BERTscore between premise and conclusion candidate, restricts novel content by selecting premise-similar content, ensuring validity at the expense of novelty.

Case study Table 4 shows clear differences in wording between the conclusions generated using different configurations for a hand-selected example. The conclusion generated without frame information mentions ‘a significant increase of energy supply’ vs. ‘significant amount of energy’ (reference conclusion). When informative label smoothing is active, the conclusion mentions the ‘potential to replace coal’, bringing in a novel element not mentioned in the premise. Adding the generic frame (Capacity and resources) interestingly leads to emphasizing the ‘potential to power the US’. Adding information about the issue-specific frame (energy) changes this back to talking about wave energy as a ‘major source of clean energy’. Conclusion reranking picks up specific elements of the premise e.g. (‘10 % of US energy supply’), but lacks novel elements compared to the given premise. The case study clearly shows that we can control conclusion generation in ways intended by our methods. However, it also shows that the observed impacts are subtle.

6 Conclusion

In this paper we have studied the question of how to condition the generation of argumentative conclusions from premises using a transformer-based fine-tuning approach based on pre-trained language models. We have shown the positive im-

pact of different strategies to bring the generated conclusions closer to the desired frame during inference while showing that proposing conclusions that are perceived as both valid and novel by humans is challenging, especially since these two dimensions seem to stand in a trade-off that renders their joint optimization difficult. Our results clearly show that the proposed strategies have the potential of improving either novelty or validity. In future work we aim to investigate the factors that contribute to validity and novelty in more detail. Especially we aim to understand how to control the trade-off between validity and novelty better to maximize the likelihood of generating conclusions that fulfill both criteria.

Acknowledgements

We are grateful to the anonymous reviewers for their valuable comments. This work has been funded by the DFG through the project ACCEPT as part of the Priority Program “Robust Argumentation Machines” (SPP1999).

References

- Yamen Ajjour, Milad Alshomary, Henning Wachsmuth, and Benno Stein. 2019. [Modeling frames in argumentation](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2915–2925, Hong Kong, China. Association for Computational Linguistics.
- Milad Alshomary, Shahbaz Syed, Martin Potthast, and Henning Wachsmuth. 2020. [Target inference in argument conclusion generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4334–4345, Online. Association for Computational Linguistics.
- Roy Bar-Haim, Lilach Eden, Roni Friedman, Yoav Kantor, Dan Lahav, and Noam Slonim. 2020. [From arguments to key points: Towards automatic argument summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4029–4039, Online. Association for Computational Linguistics.
- Maria Becker, Ioana Hulpus, Juri Opitz, Debjit Paul, Jonathan Kobbe, Heiner Stuckenschmidt, and Anette Frank. 2020. [Explaining arguments with background knowledge](#). *Datenbank-Spektrum*, 20(2):131–141.
- Maria Becker, Siting Liang, and Anette Frank. 2021. [Reconstructing implicit knowledge with language models](#). In *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 11–24, Online. Association for Computational Linguistics.
- Amber E. Boydston, Dallas Card, Justin Gross, Paul Resnick, and Noah A. Smith. 2014. [Tracking the development of media frames within and across policy issues](#).

- Dallas Card, Amber E. Boydston, Justin H. Gross, Philip Resnik, and Noah A. Smith. 2015. [The media frames corpus: Annotations of frames across issues](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 438–444, Beijing, China. Association for Computational Linguistics.
- Oana Cocarascu, Elena Cabrio, Serena Villata, and Francesca Toni. 2020. [Dataset independent baselines for relation prediction in argument mining](#). *Frontiers in Artificial Intelligence and Applications*, 326(Computational Models of Argument):45–52.
- Claes de Vreese. 2005. [News framing: Theory and typology](#). *Information Design Journal*, 13:51–62.
- Michael Fromm, Evgeniy Faerman, and Thomas Seidl. 2019. [Tacam: Topic and context aware argument mining](#). In *IEEE/WIC/ACM International Conference on Web Intelligence, WI '19*, page 99–106, New York, NY, USA. Association for Computing Machinery.
- Timon Gurcke, Milad Alshomary, and Henning Wachsmuth. 2021. [Assessing the sufficiency of arguments through conclusion generation](#). In *Proceedings of the 8th Workshop on Argument Mining*, pages 67–77, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Wei Chen. 2021. [Deberta: Decoding-enhanced bert with disentangled attention](#). In *2021 International Conference on Learning Representations*.
- Philipp Heinisch and Philipp Cimiano. 2021. [A multi-task approach to argument frame classification at variable granularity levels](#). *it - Information Technology*, 63(1):59–72.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. [The curious case of neural text degeneration](#). In *International Conference on Learning Representations*.
- Xinyu Hua and Lu Wang. 2020. [PAIR: Planning and iterative refinement in pre-trained transformers for long text generation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 781–793, Online. Association for Computational Linguistics.
- Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. 2015. [From word embeddings to document distances](#). In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15*, page 957–966. JMLR.org.
- Nona Naderi and Graeme Hirst. 2017. [Classifying frames at the sentence level in news articles](#). In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 536–542, Varna, Bulgaria. INCOMA Ltd.
- W Russell Neuman, Russell W Neuman, Marion R Just, and Ann N Crigler. 1992. *Common knowledge: News and the construction of political meaning*. University of Chicago Press.
- Juri Opitz, Philipp Heinisch, Philipp Wiesenbach, Philipp Cimiano, and Anette Frank. 2021. [Explainable unsupervised argument similarity rating with Abstract Meaning Representation and conclusion generation](#). In *Proceedings of the 8th Workshop on Argument Mining*, pages 24–35, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Debjit Paul, Juri Opitz, Maria Becker, Jonathan Kobbe, Graeme Hirst, and Anette Frank. 2020. [Argumentative relation classification with background knowledge](#). In *Computational Models of Argument - Proceedings of COMMA 2020, Perugia, Italy, September 4-11, 2020*, volume 326 of *Frontiers in Artificial Intelligence and Applications*, pages 319–330. IOS Press.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Pavithra Rajendran, Danushka Bollegala, and Simon Parsons. 2016. [Contextual stance classification of opinions: A step towards enthymeme reconstruction in online reviews](#). In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pages 31–39, Berlin, Germany. Association for Computational Linguistics.
- Paul Reisert, Naoya Inoue, Tatsuki Kuribayashi, and Kentaro Inui. 2018. [Feasible annotation scheme for capturing policy argument reasoning using argument templates](#). In *Proceedings of the 5th Workshop on Argument Mining*, pages 79–89, Brussels, Belgium. Association for Computational Linguistics.
- Benjamin Schiller, Johannes Daxenberger, and Iryna Gurevych. 2021. [Aspect-controlled neural argument generation](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–396, Online. Association for Computational Linguistics.
- Shahbaz Syed, Khalid Al Khatib, Milad Alshomary, Henning Wachsmuth, and Martin Potthast. 2021. [Generating informative conclusions for argumentative texts](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3482–3493, Online. Association for Computational Linguistics.
- Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. [Rethinking the inception architecture for computer vision](#). In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.
- Dietrich Trautmann, Johannes Daxenberger, Christian Stab, Hinrich Schütze, and Iryna Gurevych. 2020. [Fine-grained argument unit recognition and classification](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9048–9056.
- Henning Wachsmuth, Martin Potthast, Khalid Al-Khatib, Yamen Ajjour, Jana Puschmann, Jiani Qu, Jonas Dorsch, Viorel Morari, Janek Bevendorff, and Benno Stein. 2017a. [Building an argument search engine for the web](#). In *Proceedings of the 4th Workshop on Argument Mining*, pages 49–59, Copenhagen, Denmark. Association for Computational Linguistics.
- Henning Wachsmuth, Benno Stein, and Yamen Ajjour. 2017b. [“PageRank” for argument relevance](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1117–1127, Valencia, Spain. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

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

Wanzheng Zhu and Suma Bhat. 2020. [GRUEN for evaluating linguistic quality of generated text](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 94–108, Online. Association for Computational Linguistics.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. [A robustly optimized BERT pre-training approach with post-training](#). In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

A Appendix

A.1 Transformer-based generic frame classification model

To induce a generic frame classifier, similar to [Heinisch and Cimiano \(2021\)](#) we fine-tuned a ROBERTA-base ([Zhuang et al., 2021](#)) language model on the training section of the Media-Frames dataset ([Card et al., 2015](#)) (Section 3.1). We only considered text spans annotated by at least two annotators agreeing on the same frame. We encode the annotated text spans with ROBERTA-base and use the [CLS] head to predict a probability distribution over the 15 frame classes. The trained model obtained an accuracy of 58% on our held-out test split from the Media-Frames dataset.

A.2 Reference-less scores for our conclusion reranking approach

Below we describe a selection of different reference-less scores, which we use to help the conclusion reranker select an appropriate conclusion among several conclusion candidates. A first group of scores rate the conclusion candidate stand-alone, others measure the relation between premise and conclusion candidate, and some rely on the given frame information to rate the quality of a conclusion.

Conclusion-candidate-based scores To rate the quality of a conclusion candidate stand-alone, we measure its absolute token length as well as the number of non-stopword-tokens it contains and the ratio of non-stopwords-tokens to stopword-tokens. Further, we check for patterns that are typical for conclusions, such as *is*, *better than*, *should*, *therefore*.

Premise-&-conclusion-candidate-based scores Another way to rate the quality of a conclusion candidate is to characterize its relation to the premise. Here we take into account the relative conclusion candidate token length compared to the premise token length. Further, we measure coherence and grammaticality with the GRUEN-score ([Zhu and Bhat, 2020](#)) by concatenating the premise with the conclusion candidate. We also measure the similarity of the conclusion candidate and the premise using BERTscore⁸ ([Zhang et al., 2020](#)), using the outputted precision, recall, and the F1-score.

Finally, we aim to assess the argumentative relation of the conclusion candidate and the premise, by building a model that classifies this relation into attack, no relation or support. To this end, we fine-tuned a DEBERTA-base language model ([He et al., 2021](#)) for natural language inference (NLI) classification. We build SEP-structured inputs consisting of topic, premise and conclusion candidates, using the argument dataset by [Ajjour et al. \(2019\)](#) with the same train and validation split as used in our presented evaluation (Section 5). From this argument dataset we obtain positive samples (entailment/ support class) by concatenating premises with their reference conclusion. To generate samples with the neutral target class "no relation", we provide premises with conclusions sampled from other dataset topics. To provide negative samples (contradiction / attack), we join each premise with a sampled conclusion from the same topic fulfilling the same issue-specific frame but having the opposite stance towards the topic. This information is provided by the dataset. In this way, we generated a balanced dataset from [Ajjour et al. \(2019\)](#)'s dataset. The model trained on this data reaches an accuracy of 86% on the test split (including the Ω -optimization split). Using this fine-tuned language model, we tag each conclusion candidate with the predicted entailment class

⁸using the 18th layer of microsoft/deberta-xlarge without an idf-weighting

probability and a score $P(\textit{Entailment}) \cdot (1 - P(\textit{Contradiction})^2)$ to measure the risk of having a conclusion candidate which is attacked by its premise and therefore, not a valid conclusion.

Frame-sensitive scores If the input provides the issue-specific frame, we score the probability of the conclusion candidate belonging to this frame using a ROBERTA-base (Zhuang et al., 2021) language model with [CLS] conclusion candidate [SEP] issue-specific frame label [SEP] as input. The predicted value between 0 (not frame-related) and 1 (frame-related) is considered for the conclusion candidate selection. For fine-tuning such a model, we use the same train and development splits from the argument dataset of Ajjour et al. (2019) as above. We model the task as a regression task, assigning 1 for the edge case of a true issue-specific frame label and 0 for the edge case of a completely unrelated issue-specific frame label. For each positive sample that combines a conclusion with its ground truth frame label $refF$, we generate a negative sample by combining the conclusion with a frame label having the largest Word-Movers-Distance WMD (Kusner et al., 2015) $negRefF$ to the reference frame label. To have a more fine-grained regression objective, we create additionally mixed samples by randomly sampling a frame label $randF$, having a ground-truth-score of $1 - \frac{WMD(refF, randF)}{WMD(refF, negRefF)}$. The resulting mean absolute error is 0.11 on the test split (including the Ω -optimization split).

In cases where the input contains generic Media-Frames information, we take into account the probability of matching that frame, using again a fine-tuned ROBERTA-model. We use the mode described in A.1.

A.3 Further insights into the manual study

We performed an extensive manual annotation study to assess the quality of the generated conclusions for the various settings.

A.3.1 Annotators and agreement

Our aim was to collect high-quality annotations. To this end, we accepted only paid students with higher education entrance qualification working on research projects related to argument mining. After qualifying questions related to the annotation study, including a positive and negative annotation example, each student annotated indepen-

dently from the other.

Each sample was annotated three times. We split our annotation study into three rounds. The first round aims to find the best input frame configuration. In the second round, we explore the informative label smoothing. The third round rates conclusions generated using the conclusion reranking technique. The same 30 samples from the test split were used for all rounds, and all were evaluated by the same three annotators. In addition, the third round included a second bulk of 30 arguments to increase the statistic relevance. We invited two additional annotators to annotate the second bulk (keeping 1 of the previous annotators). Hence, five annotators participated in the annotation study in total.

The Fleiss-kappa-inter-annotator-agreements for the absolute judgments (yes/no) are 0.53 for validity, 0.22 for novelty and 0.4 for framing-relatedness. Among the absolute judgments, 6%, 4%, and 4% were undecided ("I don't know") for validity, novelty, and framing-relatedness, respectively. The moderate agreement for validity is relatively high for such an argumentative task. However, in general, the agreement on similar tasks has been shown to be quite low because of subjectivity (Gurcke et al., 2021). One source of this subjectivity is in the decision of where to draw the line between two categories (e.g., novel vs. not-novel).

The Fleiss-kappa-inter-annotator-agreements for the pairwise setting (Conclusion 1 is better/equal/ Conclusion 2 is better) are 0.48 for validity, 0.36 for novelty and 0.41 for framing-relatedness.

A.3.2 Annotation interface

To give further insight about the annotation task and provided instructions, Figure 4 shows a screenshot of the annotation interface in the pairwise setting, using a dummy sample. The different formatting styles (colors, borders and font style) of the conclusion boxes result from the selected rating. These interactive styles support the annotators by visualize their rating.

A.4 Analysing the gap between higher BERTscores and lower manual ratings

To provide a better understanding of the discrepancy between BERTscores and manual ratings especially in the case of activated conclusion reranking (in combination with informative label smoothing), we list a few samples in 5.

Sample X

Topic title

Premise: The text of the premise

Conclusion 1: The text of conclusion 1

Conclusion 2: The text of conclusion 2

Let's rate ;)

Validity: Conclusion is justified based on the premise

Conclusion 1

- yes
- I can't decide
- no

Conclusion 1 vs. Conclusion 2

- Conclusion 1 is more valid
- Both are equally bad/ good
- Conclusion 2 is more valid

Conclusion 2

- yes
- I can't decide
- no

Novelty: Conclusion introduces premise-related novel content (is, e.g., not a paraphrased repetition of (a part of) the premise)

Conclusion 1

- yes
- I can't decide
- no

Conclusion 1 vs. Conclusion 2

- Conclusion 1 contains more novel (proper) content
- Both contain the equal amount
- Conclusion 2 contains more novel (proper) content

Conclusion 2

- yes
- I can't decide
- no

Specific perspective »frame«

The conclusion is directed towards the perspective »frame«.

Conclusion 1

- yes
- I can't decide
- no

Conclusion 1 vs. Conclusion 2

- Conclusion 1 fits better
- Both fit equally bad/ good
- Conclusion 2 fits better

Conclusion 2

- yes
- I can't decide
- no

Any questions/ comments to this sample? (optional)

>>> Save & next >>>

Figure 4: A screenshot of the used annotation tool

Topic+issue-specific frame+Premise \mapsto <i>reference conclusion</i>	w/o conclusion reranking	conclusion rerank	rating for concl. rerank
Withdrawing from Iraq <i>political progress</i> Although the constitution and formation of a legitimate government are major achievements, there is still much work to do. Sunni Arabs have to be convinced that the new settlement is in their interests and be drawn into government. This requires that they, and others, are convinced that the Iraqi government will stand the test of time, which requires the support of the US both politically and militarily \mapsto <i>Political progress has been made but must be continued with coalition support.</i>	Sunni Arabs have to be convinced that the Iraqi government is legitimate	Iraqi Sunni Arabs must be convinced that a legitimate government exists.	validity: - -, novelty: - -, issue-specific-frame-relatedness: -
Withdrawing from Iraq <i>soldiers</i> It is odd to argue that a rapid withdraw would be risky to the lives of troops. Keeping them in Iraq is obviously risky, and particularly if it is extended over many years. \mapsto <i>Keeping troops in Iraq is obviously more dangerous than a rapid withdrawal.</i>	Withdrawing from Iraq would not risk the lives of troops	Withdrawing from Iraq would risk the lives of troops	validity: - - - -, novelty: - -, issue-specific-frame-relatedness: -
Withdrawing from Iraq <i>iraqi will</i> Sheikh Mahmood Ejemi, head of the Ejmani tribe in Hiyt, believes the improved security could quickly unravel without sufficient US support. - ""I advise the Americans to withdraw only when Iraqis can secure and achieve security and have a strong and capable military force to protect the borders and the populace. We need to have a national government that protects Iraqis, not works to isolate and kill them, like it is doing now. We need US support in fighting sectarian militias and al-Qaeda" \mapsto <i>Many Iraqis support the continued presence of US forces</i>	The US should not withdraw before the Iraqi government is sufficiently secure	Iraqis will not accept US withdrawal until they have a strong military.	validity: - - -, novelty: - -, issue-specific-frame-relatedness: ++
Vegetarianism <i>health</i> Almost all dangerous types of food-poisoning (e.g. E-coli, salmonella) are passed on through meat or eggs. Close contact between humans and animals also leads to zoonosis – diseases such as bird ‘flu which can be passed on from animals to humans. Hunters eating apes and monkeys is thought to have brought HIV/AIDS to humans. And using animal brains in the processed feed for livestock led to BSE in cattle and to CJD in humans who ate beef" \mapsto <i>Meat-eating is linked to a range of serious illness such as food-poisoning.</i>	Vegetarians are not immune to diseases of animals	Vegetarians are vulnerable to food poisoning.	validity: -, novelty: - -, frame-relatedness: -
Video surveillance <i>privacy</i> : It is certainly not the case that people monitor all security cameras closely 24/7. Most surveillance tapes are rarely seen. Usually surveillance cameras are only viewed if they have filmed a crime and are viewed only to catch criminals, not to invade people’s privacy or stalk people." \mapsto <i>Surveillance cameras are not closely monitored and are only usually viewed if a crime has taken place.</i>	Privacy infringements are rare with surveillance cameras	Surveillance cameras are rarely viewed to catch criminals.	validity: - - -, novelty: - -, issue-specific-frame-relatedness: -

Table 5: Examples of generated conclusions in which the frame-insensitive conclusion reranking technique clearly leads to better BERTscores (covering more parts of the reference conclusion) than the conclusion without reranking but receiving worse scores in the manual evaluation. Each - reflects a dispreference to the conclusion-reranking-output, while each + represents a preference rating.

LAF_T: Cross-lingual Transfer for Text Generation by Language-Agnostic Finetuning

Xianze Wu^{1*} Zaixiang Zheng² Hao Zhou^{3†} and Yong Yu^{1‡}

¹Shanghai Jiao Tong University ²Bytedance AI Lab

³Institute for AI Industry Research, Tsinghua University

{xzwu, yyu}@apex.sjtu.edu.cn

zhengzaixiang@bytedance.com

zhouhao@air.tsinghua.edu.cn

Abstract

Multilingual language pretraining enables possibilities of transferring task knowledge learned from a rich-resource source language to the other, particularly favoring those low-resource languages with few or no task annotated data. However, knowledge about language and tasks encoded is strongly entangled in multilingual neural representations, thereby the learned task knowledge falsely correlated to the source language, falling short of cross-lingual transferability. In this paper, we present a novel *language-agnostic finetuning* (LAF_T) to facilitate zero-resource cross-lingual transfer for text generation. LAF_T performs *language-agnostic task acquisition* to isolate task learning completely from the source language, and then *language specification* for better generation for specified languages. Experiments demonstrate that the proposed approach facilitates a better and parameter-efficient transferability on two text generation tasks.

1 Introduction

Deep learning has boosted the development of natural language generation (NLG), giving rise to its applications to a broad range of tasks (Brown et al., 2020; Liu et al., 2020; Xue et al., 2021), e.g., summarizing a lengthy news article. Annotated data is essential for learning neural NLG models. However, the vast bulk of available data is normally presented in English, making data scarcity in other languages a significant difficulty. Therefore, cross-lingual transfer, the ability to transfer knowledge learned in a rich-resource source language (typically English) to other, unseen target languages, has enormous practical significance.

The recent success of multi-lingual pre-trained language models (MPLMs) (Liu et al., 2020; Con-

neau et al., 2020; Xue et al., 2021) enables possibilities for such zero-resource cross-lingual transfer in a “pretraining-finetuning” paradigm. Specifically, thanks to that MPLMs can learn plausible multilingual representations for any languages involved in multi-lingual pretraining, finetuning a MPLM on task annotated data in English can exhibit immediate task performance on other languages. However, despite its appealing results on natural language understanding, the transferring performance remains unsatisfactory on language generation tasks.

The neural NLG pipeline consists of three sequential steps: a) understanding input text (e.g., a news article), b) manipulating semantics in accordance with the task (e.g., filtering out redundant content while retaining content of the main idea), and c) generating text result (e.g., abstractive summary). As a result, we suggest that learning a generation task essentially boils down to learning how to manipulate the input semantic for the following generation. However, due to the highly entangled nature of semantic information and language information learned in multilingual representations, knowledge of a downstream task learned by finetuning would inevitably be correlated to the source language, thus harming the ability to transfer to unseen target languages.

In this paper, we propose the *language-agnostic finetuning* (LAF_T). The key idea is to completely isolate acquiring task knowledge for an MPLM from the source language, and then add the language information back for generation. Given a text generation task and its annotated data in the source language, LAF_T consists of two stages:

- **Language-agnostic task acquisition.** An extra task module is added to the MPLM. The module learns to manipulate semantic content given the task without considering any information about the source language.
- **Language specialization.** We then incorporate language information back into the

*Work was done during Xianze’s internship at ByteDance AI Lab, mentored by Zaixiang.

†Work was done while at ByteDance.

‡Corresponding authors.

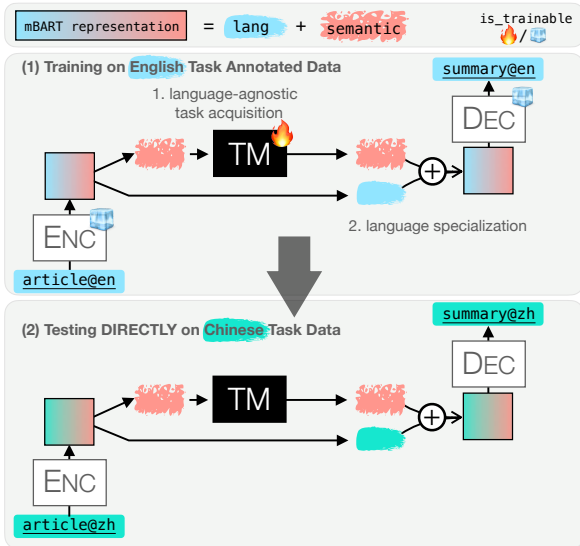


Figure 1: Illustration of LAF T for mBART. (1) Training on source language task annotated data. (2) Trained model can be directly evaluated for target language.

task module’s language-agnostic representation, helping the decoder to better generate the resulting content in the specified language.

We evaluate our zero-resource cross-lingual transfer approach in two scenarios: zero-shot and translate-train, which differ in terms of the existence of machine translation systems. Experimental results show that the proposed method facilitates a better and parameter-efficient transferability on abstractive summarization (+up to 0.71 ROUGE-L) and question generation (+up to 2.45 ROUGE-L), which could motivate further research that cross-lingual transfer necessitates careful consideration of task acquisition and language specialization,

2 Related Work

Most previous cross-lingual transfer research has succeeded on NLU rather than NLG. For both NLU and NLG, one solution is data augmentation that leverages data from the source language to the target language using translation systems or code-switching (Singh et al., 2019; Bornea et al., 2021; Qin et al., 2020). Some NLU research aims to learn language-agnostic features that minimize the distance among features from different languages, by adversarial training (Keung et al., 2019; Chen et al., 2019), removing the language identity from the original multi-lingual representations (Libovický et al., 2020; Zhao et al., 2021; Yang et al., 2021; Tiyajamorn et al., 2021) or contrastive learning (Yu and Joty, 2021).

For NLG, one of the most promising findings of cross-lingual transfer is that multilingual machine translation systems trained on massive amount of multilingual data manifest emergent ability of unsupervised (Üstün et al., 2021) or zero-shot translation for those unseen language pairs (Gu et al., 2019; Chen et al., 2022). Such observations encourage researchers to design effective pretraining objectives favoring cross-lingual transfer for monolingual text generation tasks (e.g., summarization) (Chi et al., 2020; Lewis et al., 2020; Maurya et al., 2021), whereas the finetuning process receives little attention. Despite learning language-agnostic features for finetuning as in NLU is promising, language information, in contrast to NLU, is critical for NLG. If only language-agnostic features are used, the model will not be able to generate text in the specified language.

3 Methodology: LAF T

Figure 1 shows the overall workflow of LAF T when applying to mBART (Liu et al., 2020).¹ As illustrated, we first introduce an extra task module (TM), parameterized by two Transformer layers (Vaswani et al., 2017), between the encoder and decoder for **language-agnostic task acquisition** (§3.1), where the TM is expected to learn how to manipulate input semantic content given the task. We then perform **language specialization** by adding language information to the language-agnostic representation obtained by the TM, allowing the decoder to synthesize the resulting text in the provided language (§3.2).

3.1 Language-agnostic Task Acquisition

Our approach is inspired by Yang et al. (2021) that for an MPLM, the representations from the same language L tend to cluster together, which implies that they share vector space components that correspond to the language identity of the language L . This finding intuitively enables disentangling the semantic contents from language identity by removing the language components from the representation, which can be conducted as following two steps:

(1) Estimation of language component. Given a pretrained mBART, its encoder can be seen as a multi-lingual embedding system E . For

¹In this paper, we primarily study the proposed language-agnostic finetuning on mBART, but the method can be applied to any encoder-decoder MPLMs.

each language L , we construct a language matrix $M_L \in \mathbb{R}^{n \times d}$ based on a collection of monolingual texts $\{t_L^i\}_{i=0}^n$, where the i th row of M_L is the sentence representation of t_L^i given by E . We then apply singular value decomposition (SVD) $M_L = U_L \Sigma_L V_L^T$, and extract the first k right singular vectors (i.e., columns of $V_L \in \mathbb{R}^{d \times d}$) as the shared components for language identity of L , denoted as $c_L \in \mathbb{R}^{d \times k}$.

(2) Removal of language component. Given a text $x_L = \{x_L^i\}$ from the language L , where x_L^i is the i th token of x_L , we denote the representation of x_L^i given by the encoder as e_L^i . The sentence representation e_L is obtained via the mean-pooling of $\{e_L^i\}$. Then we subtract the projection of e_L onto c_L from e_L^i as

$$r_L^i = e_L^i - c_L \frac{c_L^T e_L}{\|e_L\|_2}.$$

As a result, $r_L = \{r_L^i\}$ is the language-agnostic representation as expected, which is then fed into the TM for learning the task:

$$h_L = \text{TM}(r_L)$$

3.2 Language Specialization for Generation

The proposed language-agnostic task acquisition eases the transfer of task knowledge across language. Unlike NLU tasks, which can rely solely on semantic information for classification, language information is critical for NLG tasks since we want to generate text in a specific language. Thus, beside language-agnostic task acquisition, we also need to improve the model regarding its language generation ability. We refer to this as language specialization, which includes two aspects: (1) we integrate the subtracted language components into the TM’s language-agnostic output, (2) we enhance the decoder with an extra language adapter.

Fusing with subtracted language components. We apply a fusion mechanism to add subtracted language components c_L back to the TM’s output:

$$\mathbf{B}(h_L^i, c_L) = \mathbf{U}(\text{ReLU}(\mathbf{D}([h_L^i, c_L]))) + h_L^i,$$

where $\mathbf{D} \in \mathbb{R}^{2d_h \times d_a}$ and $\mathbf{U} \in \mathbb{R}^{d_a \times d_h}$ are parametrized by two feed-forward layers. $\mathbf{B}(h_L^i, c_L)$ is then fed into the decoder.

Enhancing decoder with language adapter. The decoder is responsible for generating text in a given language. To promote the decoder to adapt to

the fused representations, we incorporate a feed-forward layer based language adapter to each decoder layer (Pfeiffer et al., 2020a), which is jointly trained with the fusion mechanism.

3.3 Learning

Learning of LAFT contains two stages.

- (1) *Unsupervised generation pretraining.* In this stage, we only allow the TM and fusion mechanism trainable while keeping the remainder of the model parameters frozen. We leverage *unsupervised data* from the source and target language. Following (Liu et al., 2020), we use a cross-entropy loss between the original document and the decoder’s output given the corrupted document as input, which is constructed by applying “text infilling” noise to the original document (Liu et al., 2020).
- (2) *Task finetuning.* In this stage, given *source language annotated task data*, we freeze the fusion mechanism and optimize the TM using the cross-entropy loss between the decoder’s output and the ground-truth reference.

4 Experiments

We experiment on two NLG tasks, i.e., abstractive text summarization and question generation to evaluate our LAFT for cross-lingual transfer.

Datasets. For text summarization, we perform experiments on the XGIGA datasets. We choose its English part as the training set and its French and Chinese parts as the evaluation set. For question generation, we choose the XQG dataset (Chi et al., 2020). The XQG dataset consists of the English part and the Chinese part. We train models on English part and evaluate models on Chinese part.

We learn language specialization using cc100 dataset (Conneau et al., 2020), from which we select a subset containing 1,000,000 sentences for Chinese, English and French respectively.

Baselines. We compare LAFT with the following baselines:

- mBART (full): directly finetuning the full parameters of mBART on English annotated data;
- mBART (enc): only finetuning the encoder parameters of mBART;
- TM + adv: using adversarial training instead of LAFT to force the output of TM to be language-agnostic.

More details are presented in Appendix.

Setting	Zero-shot		Trans-train	
	zh→zh	fr→fr	zh→zh	fr→fr
Baselines				
mBART (full)	43.82	33.40	47.33	42.8
mBART (enc)	45.85	36.55	47.09	42.11
TM + adv	31.41	36.71	48.04	43.04
LAFt	46.37	40.78	47.66	43.10

Table 1: Results of abstractive summarization. “full”: finetuning full model. “enc”: finetuning only encoder

Setting	Zero-shot	Trans-train
Language	zh→zh	zh→zh
Baselines		
mBART (full)	21.62	36.58
mBART (enc)	32.08	33.57
TM + adv	21.98	37.02
LAFt	34.53	37.02

Table 2: Results of question generation. “full”: finetuning full model. “enc”: finetuning only encoder.

Results of Zero-shot Setting. First, we evaluate models on the zero-shot cross-lingual transfer. Results of abstractive summarization and question generation are presented in Table 1 and Table 2, respectively. When a full mBART is fine-tuned, it runs the danger of incorrectly associating the task to the source language, resulting in poor transfer performance. Only Finetuning the encoder can somehow alleviate but does not fundamentally address the problem. LAFt, on the other hand, can learn task ability avoiding associating to the source language, which improves transferability for generation and outperforms baseline systems. Surprisingly, while the adversarial method is known to be good at removing language information, it fails miserably in the zero-shot case due to a lack of task data for each language, causing the model to degenerate into copying the input sequence regardless of languages.

Results of Translate-train Setting. We evaluate models on the translate-train setting to see if data augmentation by machine translation could further help. As shown in Table 1 and Table 2, we can observe that data augmentation can generally improve all approaches. Note that because pseudo task data for target languages is accessible in this setting, the adversarial method can function normally. Nevertheless, our LAFt still achieves comparable results with the adversarial method, demonstrating the effectiveness of the proposed method.

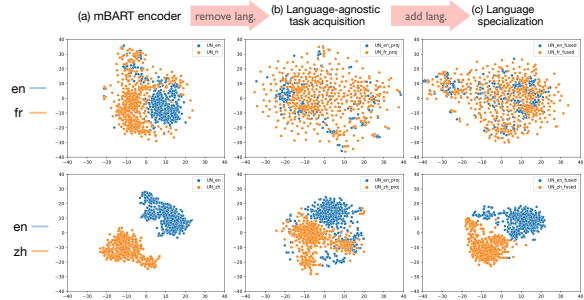


Figure 2: t-SNE (Van der Maaten and Hinton, 2008) visualization of representations.

Model	R-L (\uparrow)	$ \theta_{\text{trainable}} $ % (\downarrow)
mBART (full)	43.82	100%
mBART (enc)	45.85	19.2%
mBART (enc top-2)	<u>44.85</u>	<u>3.8%</u>
Adapter (Pfeiffer et al., 2020a)	<u>43.05</u>	<u>4.3%</u>
LAFt	46.37	3.8%

Table 3: Number of trained parameters and results on abstractive summarization. “enc top-2”: only finetuning the top two layers of the encoder.

Visualization of LAFt. To ensure that LAFt can yield language-agnostic representations, we visualize the representations before and after applying LAFt in Figure 2. As we can see, the original mBART encoder representation is distributed separately in terms of languages (Figure 2(a)). After removing language identity, the distribution of representations from different languages becomes closer, allowing the model to produce language-agnostic representations for task acquisition (Figure 2(b)). Finally, once language specialization is performed, the representations become language-aware thus distribute separately again, making it easier for the decoder to generate text in a specific language (Figure 2(c)).

Analysis of Parameter Efficiency. To demonstrate parameter efficiency of LAFt, we compare the performance of abstractive summarization with the number of training parameters. As shown in Table 3, our method yields the best ROUGE-L score with the fewest training parameters, demonstrating that LAFt results in a parameter-efficient model.

5 Conclusion

This paper proposes language-agnostic finetuning (LAFt) to facilitate zero-resource cross-lingual transfer for text generation. We finetune a task module only through the semantic contents of a multi-lingual representation. To achieve it, we utilize a disentangled-based and an adversarial-based

method. Then we combine the information of a language with the task module’s language-agnostic representation, allowing the model to generate text in the language. Experimental results show that language-agnostic finetuning results in a better and parameter-efficient transferability on two text generation tasks. The major limitation of our work is we only explore two target languages. We leave other languages for future work.

6 Acknowledgements

We would like to thank the anonymous reviewers for their valuable comments.

References

- Mihaela A. Bornea, Lin Pan, Sara Rosenthal, Radu Florian, and Avirup Sil. 2021. [Multilingual transfer learning for QA using translation as data augmentation](#). In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 12583–12591. AAAI Press.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). *CoRR*, abs/2005.14165.
- Guanhua Chen, Shuming Ma, Yun Chen, Dongdong Zhang, Jia Pan, Wenping Wang, and Furu Wei. 2022. [Towards making the most of cross-lingual transfer for zero-shot neural machine translation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 142–157. Association for Computational Linguistics.
- Xilun Chen, Ahmed Hassan Awadallah, Hany Hassan, Wei Wang, and Claire Cardie. 2019. [Multi-source cross-lingual model transfer: Learning what to share](#). In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3098–3112. Association for Computational Linguistics.
- Zewen Chi, Li Dong, Furu Wei, Wenhui Wang, Xian-Ling Mao, and Heyan Huang. 2020. [Cross-lingual natural language generation via pre-training](#). In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7570–7577. AAAI Press.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.
- Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O. K. Li. 2019. [Improved zero-shot neural machine translation via ignoring spurious correlations](#). In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1258–1268. Association for Computational Linguistics.
- Phillip Keung, Yichao Lu, and Vikas Bhardwaj. 2019. [Adversarial learning with contextual embeddings for zero-resource cross-lingual classification and NER](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1355–1360. Association for Computational Linguistics.
- Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida I. Wang, and Luke Zettlemoyer. 2020. [Pre-training via paraphrasing](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Jindrich Libovický, Rudolf Rosa, and Alexander Fraser. 2020. [On the language neutrality of pre-trained multilingual representations](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1663–1674. Association for Computational Linguistics.
- Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. [Multilingual denoising pre-training for neural machine translation](#). *Trans. Assoc. Comput. Linguistics*, 8:726–742.
- Kaushal Kumar Maurya, Maunendra Sankar Desarkar, Yoshinobu Kano, and Kumari Deepshikha. 2021. [Zmbart: An unsupervised cross-lingual transfer framework for language generation](#). In *Findings of the Association for Computational Linguistics*:

- ACL/IJCNLP 2021, Online Event, August 1-6, 2021, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 2804–2818. Association for Computational Linguistics.
- Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulic, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020a. [Adapterhub: A framework for adapting transformers](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 46–54. Association for Computational Linguistics.
- Jonas Pfeiffer, Ivan Vulic, Iryna Gurevych, and Sebastian Ruder. 2020b. [MAD-X: an adapter-based framework for multi-task cross-lingual transfer](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7654–7673. Association for Computational Linguistics.
- Libo Qin, Minheng Ni, Yue Zhang, and Wanxiang Che. 2020. [Cosda-ml: Multi-lingual code-switching data augmentation for zero-shot cross-lingual NLP](#). In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3853–3860. ijcai.org.
- Jasdeep Singh, Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2019. [XLDA: cross-lingual data augmentation for natural language inference and question answering](#). *CoRR*, abs/1905.11471.
- Nattapong Tiyajamorn, Tomoyuki Kajiwara, Yuki Arase, and Makoto Onizuka. 2021. Language-agnostic representation from multilingual sentence encoders for cross-lingual similarity estimation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7764–7774.
- Ahmet Üstün, Alexandre Berard, Laurent Besacier, and Matthias Gallé. 2021. [Multilingual unsupervised neural machine translation with denoising adapters](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6650–6662. Association for Computational Linguistics.
- Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. [mt5: A massively multilingual pre-trained text-to-text transformer](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 483–498. Association for Computational Linguistics.
- Ziyi Yang, Yinfei Yang, Daniel Cer, and Eric Darve. 2021. [A simple and effective method to eliminate the self language bias in multilingual representations](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5825–5832. Association for Computational Linguistics.
- Tao Yu and Shafiq R. Joty. 2021. [Effective fine-tuning methods for cross-lingual adaptation](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 8492–8501. Association for Computational Linguistics.
- Wei Zhao, Steffen Eger, Johannes Bjerva, and Isabelle Augenstein. 2021. [Inducing language-agnostic multilingual representations](#). In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics, *SEM 2021, Online, August 5-6, 2021*, pages 229–240. Association for Computational Linguistics.

A Appendix

Implement Details. We choose the mBART_{large} model as the backbone model. The task module consists of two transformer layers, whose setting is the same as the transformer layer in the mBART_{large} model. Language adapters are appended by each decoder layer. We follow the setting of language adapter used in (Pfeiffer et al., 2020b) while moving the layer normalization to the end of the adapter. For all experiments, we set d_a as 1024 and k as 6.

We utilize the Adam optimizer with learning rate scheduling. The warm-up step is 10000, and linear learning weight decay is used in the remaining training. We select the maximum learning rate from $\{1e - 4, 3e - 5\}$ according to the best result on the evaluation set. Decoding is done with beam search (beam size = 5) and length penalty ($\alpha = 1.5$ for text summarization and $\alpha = 3$ for question generation).

Adversarial-based method. The main idea is to use adversarial training to force the output of the

TM to be language-agnostic. Specifically, we introduce a language classifier to judge whether or not a text is from the source language. Given the TM's output h_L of a text x_L , the classifier calculates the probability that x_L belongs to the source language L_{src} as $\hat{y} = x_L \mathbf{W}_c^T$, where $\mathbf{W}_c \in \mathbb{R}^{d_a \times 1}$ is the weight of the classifier. We encourage the classifier to recognize x 's language identity by minimizing a cross-entropy:

$$\mathcal{L}_{cls} = -\mathbb{I}_{x \in L_{src}} \cdot \log(\hat{y}) - (1 - \mathbb{I}_{x \in L_{src}}) \cdot \log(1 - \hat{y}),$$

where $\mathbb{I}_{x \in L_{src}} = 1$ when x is from the source language, otherwise 0. On the other hand, we encourage the TM to fool the language classifier:

$$\mathcal{L}_{adv} = -\mathbb{I}_{x \in L_{src}} \cdot \log(1 - \hat{y}) - (1 - \mathbb{I}_{x \in L_{src}}) \cdot \log(\hat{y}).$$

Besides, we utilize the cross-entropy loss between the decoder's output and the target sequence:

$$\mathcal{L}_{gen} = -(1 - \epsilon) \log p(i) - \sum_{j \neq i \in V} \frac{\epsilon}{|V| - 1} \log p(j)$$

The final loss is,

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{adv} + \mathcal{L}_{gen}$$

Note that the adversarial training needs data from the target language. As the annotated data from the target language can not be accessed, we leverage monolingual data.

Using multi-lingual representations. Like LAFT, we also need to provide language information to TM's output. Given the TM's output h_L^i and the encoder's output e_L^i , a gated mechanism aggregates h_L^i and e_L^i via a weighted sum as

$$\begin{aligned} \alpha^i &= \text{sigmoid}(\mathbf{W}_g([h_L^i, e_L^i]) + \mathbf{b}_g) \\ g_L^i &= \alpha^i h_L^i + (1 - \alpha^i) e_L^i \end{aligned}$$

where $\mathbf{W}_g \in \mathbb{R}^{d_h + d_a}$. Unlike the fusion mechanism, the gated mechanism is trained along with the whole model.

Quantum Natural Language Generation on Near-Term Devices

Amin Karamlou*
IBM Quantum
University of Oxford

Marcel Pfaffhauser
IBM Quantum

James Wootton
IBM Quantum

Abstract

The emergence of noisy medium-scale quantum devices has led to proof-of-concept applications for quantum computing in various domains. Examples include Natural Language Processing (NLP) where sentence classification experiments have been carried out, as well as procedural generation, where tasks such as geopolitical map creation, and image manipulation have been performed. We explore applications at the intersection of these two areas by designing a hybrid quantum-classical algorithm for sentence generation.

Our algorithm is based on the well-known simulated annealing technique for combinatorial optimisation. An implementation is provided and used to demonstrate successful sentence generation on both simulated and real quantum hardware. A variant of our algorithm can also be used for music generation.

This paper aims to be self-contained, introducing all the necessary background on NLP and quantum computing along the way.

1 Introduction

It is widely believed that computers operating according to the laws of quantum mechanics will outperform classical computers at specialised tasks. This belief is backed up by the fact that important computational problems such as integer factorisation (Shor, 1997) and unstructured search (Grover, 1996) admit quantum algorithms which are provably faster than the best known classical algorithms for solving them. Unfortunately, in order to make use of these algorithms, we would first need to build scalable, fault-tolerant quantum computers, which are still some years away. By contrast, the current generation of quantum computers are still fairly rudimentary, containing at most a few hundred noisy qubits, i.e. qubits with which we cannot

perform perfect operations (Preskill, 2018). Despite their shortcomings, these devices represent a significant milestone for quantum computing. This is because unlike their smaller predecessors they cannot be simulated efficiently on classical hardware. Hence, it is possible that near-term quantum devices will bring with them the first examples of tasks performed by quantum computers that not even the most powerful classical supercomputers can perform, with tentative first steps made for proof-of-principle problems (Arute et al., 2019; Pednault et al., 2019). The search for examples in which a useful advantage can be demonstrated has led to the development of tailor-made algorithms for near-term devices that solve problems in domains such as chemistry, and optimisation (Farhi et al., 2014; Peruzzo et al., 2014).

In this paper, we are concerned with near-term quantum algorithms for natural language generation (NLG). NLG lies at the intersection of procedural generation, i.e. the algorithmic generation of data, and Natural Language Processing (NLP), both of which are active research topics within the quantum software community (see e.g. Wootton, 2020b,a; Coecke et al., 2020; Lorenz et al., 2021). The importance of NLG is underscored by its wide range of potential applications. It can for instance be used in video games to create natural-sounding dialogue, or in journalism to create automated news articles. These applications are often time-sensitive, as in the case of video games, where delays in dialogue generation would make the user experience unsatisfactory. In other situations, NLG algorithms have to deal with a large amount of input data. This is the case in automated journalism where information from many different sources needs to be collated into one coherent article. These considerations mean that developing faster algorithms for NLG tasks would have tremendous practical consequences. Thus, it is natural to wonder if any such tasks can benefit from speedups when performed

*corresponding author: Amin.Karamlou@cs.ox.ac.uk.

on a quantum computer. Our aim here is to take the first steps towards answering this question.

Throughout this work we will make use of the well-established mathematical connection between the Distributional Compositional Categorical (DisCoCat) (Coecke et al., 2010) model of natural language and quantum theory. This connection was recently exploited in several works (Meichanetzidis et al., 2020; Lorenz et al., 2021) to successfully perform Quantum Natural Language Processing (QNLP) on real quantum hardware (as opposed to simulation with conventional hardware). More specifically it was used to perform the task of binary sentence classification. The aim of this task is simple: Given a sentence about one of two possible topics, decide which topic it is about. Building upon this work, we design a sentence generation algorithm that can run on current quantum hardware. Our algorithm takes as input one of several possible topics and produces as output a sentence with that topic. Our algorithm works by searching through the space of possible sentences using simulated annealing (SA), a well-known probabilistic method for solving combinatorial optimisation problems. The choice of SA is motivated by the recent success of the method at (classically) solving the task of sentence paraphrasing (Liu et al., 2020). We experimentally evaluate the performance of our algorithm at news headline generation. We also show how our algorithm can be adapted to perform music generation.

Before continuing it is worth clarifying the goal of this paper and the scope of our claims. The formal similarity between DisCoCat and quantum theory has led to some authors claiming that NLP is an inherently “quantum native“ field (Coecke et al., 2020), and that we can expect large-scale quantum computational speedups for NLP tasks as more powerful quantum hardware becomes available. Testing these claims theoretically would require significant analysis of QNLP proposals using computational complexity theory, as has been done with other proposals for quantum advantage, for example in Aaronson and Chen (2016); Brakerski et al. (2020); Zhu et al. (2021). Alternatively, we could wait for larger quantum computers to be built, allowing for experimental comparison of QNLP algorithms and cutting edge classical methods such as GPT-3 (Brown et al., 2020) or BERT (Devlin et al., 2019). We do not claim to address either one of these challenges here. Our work is rather a

proof-of-concept example of how NLG can be performed on quantum hardware. We also hope that by assuming a modest mathematical background this paper can serve as an introduction to quantum software design using the diagrammatic style of quantum theory utilised in QNLP research.

The rest of the paper is organised as follows: In section 2 we describe the necessary background on DisCoCat and quantum computing. Section 3 contains the details of our SA based sentence generation algorithm. We report the results of experiments with this algorithm in section 4, including a discussion of how the algorithm can be adapted for music composition in section 3.3. Finally, we discuss future research avenues in section 5.

2 Preliminaries

2.1 Quantum Computing

This section presents a self-contained overview of the basics of quantum computation, assuming no familiarity with the topic. Naturally, what we present is far from a complete introduction. A more in-depth book for further reading is Nielsen and Chuang (2002). Alternatively, Coecke and Kissinger (2018) introduces quantum theory via the diagrammatic language used here.

The idea behind quantum computation is to harness features of quantum mechanics that have no classical analogue in the design of efficient algorithms. The first of these features worth mentioning is called *superposition*. The logical building blocks of a classical computer are bits. These are objects that can have one of two possible states, 0 or 1. The quantum analogue of a bit, known as a qubit, has a state that lives in a 2-dimensional Hilbert space. We use the notation¹ $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ to denote the orthonormal basis vectors of this space. The state of a qubit, written as $|\psi\rangle$, is a linear combination of these basis vectors:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle \text{ s.t. } \alpha, \beta \in \mathbb{C}, |\alpha|^2 + |\beta|^2 = 1$$

It is this linear combination that is referred to as a superposition.

The act of reading the value of a qubit in state $|\psi\rangle$ is called a *measurement*. Regardless of what

¹This is referred to as Dirac or bra-ket notation and is used ubiquitously throughout quantum information. See appendix 10 of (De Wolf, 2019) for a concise introduction to this formalism.

superposition a qubit is in, the result of a measurement is always one of two possible outcomes, 0 or 1. The probability of measuring 0 is equal to $|\alpha|^2$, and α is known as the *amplitude* of $|0\rangle$. Likewise, the probability of measuring 1 is $|\beta|^2$, and β is known as the *amplitude* of $|1\rangle$. Crucially, once a measurement has occurred, the state $|\psi\rangle$ *collapses* to the corresponding basis state. For example, if we measure a qubit in state $|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ and observe the outcome 0, then immediately after the measurement the state of the qubit is $|0\rangle$.

Naturally, to perform a meaningful computation we need to use more than just one qubit. The *joint state* $|\phi\rangle$ of n qubits lives in a Hilbert space of dimension $N = 2^n$ with orthogonal basis states of the form $|b_1\rangle \otimes |b_2\rangle \otimes \dots \otimes |b_n\rangle$ where each $b_i \in \{0, 1\}$. We will abbreviate these basis states to $|b_1b_2b_3\dots b_n\rangle$. With some abuse of notation it will also often be convenient to write these basis states in decimal notation i.e. $|0\rangle = |000\dots000\rangle$, $|1\rangle = |000\dots001\rangle$, $|2\rangle = |000\dots010\rangle$, ... $|N-1\rangle = |111\dots111\rangle$.

$|\phi\rangle$ is then once again a superposition:

$$|\phi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle + \dots + \alpha_{N-1}|N-1\rangle$$

$$\text{s.t. } \forall i \alpha_i, \beta \in \mathbb{C}, \sum_i |\alpha_i|^2 = 1$$

When measuring $|\phi\rangle$ one observes outcome i with probability $|\alpha_i|^2$ and the state of the underlying qubits collapses to $|i\rangle$.

Aside from measurement, a quantum system can also be manipulated using *quantum logic gates*. Mathematically, these gates are unitary linear maps U . Thus, the evolution of a system from one timestamp to the next can simply be described as $|\psi_1\rangle = U|\psi_0\rangle$.

Pictorially, a quantum computation can be represented as a circuit. Figure 1 provides an example of such a circuit. In this example, two qubits begin in the joint state $|\psi_0\rangle = |0\rangle$. A quantum logic gate $H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, known as a hadamard gate is applied to each qubit, transforming the state into $|\psi_1\rangle = H \otimes H |0\rangle = \frac{1}{2}|0\rangle + \frac{1}{2}|1\rangle + \frac{1}{2}|2\rangle + \frac{1}{2}|3\rangle$. Finally, the state is measured, resulting in one of the four possible outputs 0, 1, 2, or 3 being observed, each with a probability of $\frac{1}{4}$. After measurement, the state collapses to the respective basis state $|0\rangle$, $|1\rangle$, $|2\rangle$, or $|3\rangle$.

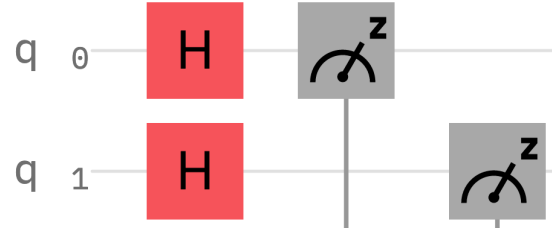


Figure 1: A simple quantum circuit created using the IBM Quantum Composer available at <https://quantum-computing.ibm.com/>.

2.2 DisCoCat and QNLP

The Distributional Compositional Categorical (DisCoCat) model of language meaning (Coecke et al., 2010) is a mathematical framework that allows for the meaning of a sentence to be described as a combination of the meaning of its constituent words, and the grammatical relationships between these words. This is in contrast to many older NLP models, which treat sentences as “bags of words” while ignoring their grammatical structure.

DisCoCat comes equipped with a pictorial representation, allowing any sentence to be represented by a so-called *string diagram*. Such a diagram consists of boxes representing words, and wires connecting these boxes according to the formalism of pregroup grammars (Lambek, 2008). This means that every wire in the diagram is annotated either by some atomic type p , a left adjoint $p.l$, or a right adjoint $p.r$. Let us explain the role of types and adjoints through example, by considering the sentence “Alice generates language“. The DisCoCat diagram corresponding to this sentence is given in figure 2. In this diagram, wires are annotated by the noun type n and the sentence type s . As we can see, the box for the word ‘generates’ has three wires coming out of it, which are annotated by $n.r$, s , and $n.l$ respectively. This indicates that the word ‘generates’ expects to receive a noun on its left (in this case ‘Alice’), as well as another noun on its right (in this case ‘language’) in order to output a grammatical sentence. In general, a sentence is grammatical if its DisCoCat diagram has a single open output wire of type s , as in the example of figure 2.

It is worth noting that DisCoCat diagrams are more than simple pictures. They are based on the rigorous formalism of monoidal categories (Heunen and Vicary, 2019, Chapter 1), which means they are equipped with a diagrammatic calculus. This calculus can be used to rewrite complicated

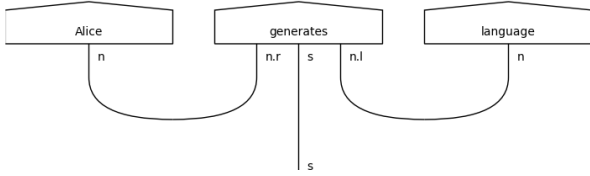


Figure 2: DisCoCat diagram for the sentence ‘Alice generates language.’

string diagrams into simpler ones that still encode the meaning of the original sentence. As it happens, monoidal categories and string diagrams also turn out to be a suitable high-level framework for capturing much of quantum information and computation (Abramsky and Coecke, 2004; Coecke and Kissinger, 2018). This observation is part of the reason that one may hope for quantum advantage in NLP tasks in the long term.

We now outline a procedure for transforming any sentence into a parameterised quantum circuit that can be run on real IBM Quantum hardware. The pipeline we discuss here has recently been implemented as part of *lambeq* (Kartsaklis et al., 2021), a python library developed specifically for QNLP tasks.

1. A sentence is converted to a DisCoCat diagram using the Combinatory Categorical Grammar (CCG) based techniques of Yeung and Kartsaklis (2021).
2. The DisCoCat diagram is simplified using some of the rewrite rules available in *lambeq*. Even though this step is strictly speaking optional, applying rewrite rules often leads to crucial computational advantages, for instance by reducing the number of qubits required to implement the parameterised quantum circuit.
3. An *ansatz* is used to transform the simplified diagram to a parameterised quantum circuit. This *ansatz* is a mapping that assigns a number of qubits to each wire type in the string diagram, as well as a set of quantum logic gates to each word in the diagram.
4. The quantum compiler *t|ket* (Sivarajah et al., 2020) is used to translate the parameterised quantum circuit into machine-specific instructions, which can be executed on real IBM quantum computers.

In this paper we use the IQP *ansatz*. This transforms each DisCoCat diagram into an Instantaneous

Quantum Polynomial (IQP) circuit. We do not justify this choice of *ansatz* here, more information is available in (Havlíček et al., 2019; Lorenz et al., 2021). The parameterised quantum circuit corresponding to “Alice generates language” is given in figure 3.

2.3 Sentence Classification

Before we can present our sentence generation algorithm we must first explain how sentence classification can be performed on near-term quantum devices. What we outline here is a step-by-step overview for solving the following task: Given a dataset Γ of sentences, each of which belongs to one of k possible topics, train a classifier that can correctly determine the topic of further unseen sentences (provided the unseen sentences are also about one of the k possible topics). This section mostly follows Lorenz et al. (2021), although we modify the algorithm to perform multi-class rather than binary sentence classification.

1. Each sentence $S \in \Gamma$ is converted to a parameterised quantum circuit C_S using the techniques discussed in the previous section. Note that some parameters may be shared between quantum circuits corresponding to different sentences. This occurs when the same words appear in multiple sentences. We set $q_n = 1$, and $q_s = \lceil \log k \rceil$, where q_n and q_s are the number of qubits associated to the noun and sentence wire types respectively. Measuring such a circuit yields one of k possible outcomes, each of which we associated with one of the topics in our corpus.
2. For each sentence $S \in \Gamma$ and each topic $i \in \{0, 1, \dots, k - 1\}$ we define a binary predicate $L(i, S) \in \{0, 1\}$ and set $L(i, S) = 1$ if and only if sentence S has topic i . Moreover, we write $P(i, C_S)$ for the probability of observing outcome i when measuring the final state of a quantum circuit C_S . Finally, let Ω denote the full set of parameters used in all the quantum circuits combined. Our goal is thus to find the optimal Ω which maximises $P(i, C_S)$ whenever $L(i, S) = 1$. This problem can be solved using classical machine learning techniques, by minimising the categorical cross-entropy loss function below. This is achieved by using the Simultaneous perturbation stochastic approximation (SPSA) algorithm (Spall, 1998).

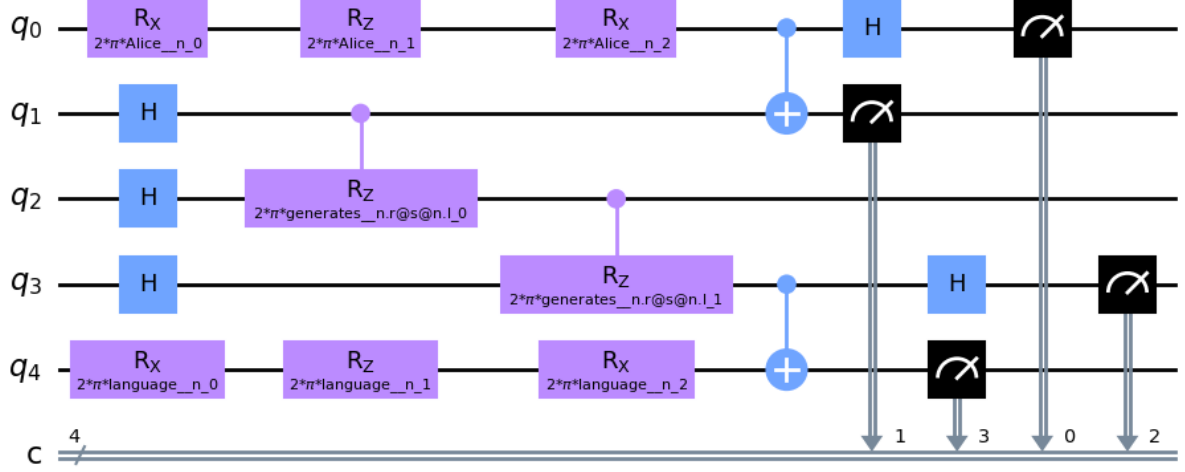


Figure 3: Parameterised quantum circuit for the sentence “Alice generates language”.

$$C(\Omega) = \sum_{S \in \Gamma} L(i, S) \cdot \log P(i, C_S)$$

- Given an unseen sentence $S \notin \Gamma$ we can now predict its topic as follows: Use the optimal parameters Ω to create the quantum circuit C_S . Measure the final state of C_S obtaining an outcome $i \in \{0, 1, \dots, k - 1\}$. Output the topic associated with outcome i .

3 Sentence Generation

In this section, we present our hybrid quantum-classical sentence generation algorithm.

We first discuss the simulated annealing (SA) algorithm for solving combinatorial optimisation problems (Kirkpatrick et al., 1983). Then, we rigorously formulate our sentence generation task as an optimisation problem and show in detail how a version of SA can be used to efficiently generate and test many candidate sentences until a satisfactory one is found.

3.1 Simulated Annealing

An optimisation problem is a problem where a satisfactory solution must be found from a search space of possible solutions. By a satisfactory solution we mean one that maximises (or comes close to maximising) some objective function over the search space.

Simulated annealing (SA) is a well-known heuristic method for solving optimisation problems. Let \mathcal{X} be a search space, and $f : \mathcal{X} \rightarrow [0, 1]$ be an objective function over that search space. The goal of SA is to find $x \in \mathcal{X}$ which maximises

$f(x)$. SA starts by either randomly or heuristically choosing a starting candidate state $x_0 \in \mathcal{X}$. At each step t , the algorithm then considers some neighbouring state x^* of the current candidate x_t . If $f(x^*) > f(x_t)$ then the algorithm ‘accepts’ x^* by setting $x_{t+1} = x^*$ and beginning a new iteration. In the event that x^* is not accepted SA simply sets $x_{t+1} = x_t$ and begins a new iteration. Even if $f(x^*) \leq f(x_t)$ SA may still accept x^* with some small probability $e^{\frac{f(x^*) - f(x_t)}{T}}$. This is known as the *metropolis* criterion and depends on an annealing temperature T . There are many different options available for calculating T at each timestep. Usually this value is set to be high at the start of SA so that x^* has a high acceptance probability. With each iteration, the value of T decreases, allowing SA to converge towards a solution. In this work, we use the *fast simulated annealing* algorithm which sets $T = \frac{T_i}{t+1}$ at each iteration, where T_i is the initial temperature.

Simulated annealing performs well in practice and is guaranteed to converge towards the optimal solution under reasonable assumptions (Granville et al., 1994). Although in the worst-case this convergence may take a prohibitively long amount of time.

3.2 The Algorithm

Let us assume that we have trained a multi-class sentence classifier using the techniques discussed in section 2.3. The sentence generation task we aim to solve is the following: Given as input one of the topics $i \in \{0, 1, \dots, k\}$ which the classifier is trained over, produce a sentence with that topic.

This task can be seen as an optimisation problem where the search space \mathcal{X} consists of all sentences formed from the vocabulary used to train the classifier². The objective function f can then simply be defined as $f(S) = P(i, C_S)$. Where C_S is the quantum circuit generated using the optimal parameters Ω . As per the discussion in section 2.3 This function is maximal whenever the sentence S has a high probability of being classified with topic i . We now outline the procedure for solving this optimisation problem using SA.

1. Start by generating a random candidate sentence s_0 from our vocabulary.
2. At each step t we generate a neighbouring state s^* of s_t . This generation proceeds similarly to the word level editing approach of Miao et al. (2019). More specifically, let $s_t = [w_1, w_2, \dots, w_n]$. s^* is generated by randomly performing one of the following editing operations:
 - *Insert*: randomly selects a word w and an index j and sets $s^* = [w_1, \dots, w_{j-1}, w, w_j, \dots, w_n]$.
 - *Delete*: randomly selects an index j and sets $s^* = [w_1, \dots, w_{j-1}, w_{j+1}, \dots, w_n]$.
 - *Replace*: randomly selects a word w and an index j and sets $s^* = [w_1, \dots, w_{j-1}, w, w_{j+1}, \dots, w_n]$.
3. Calculate the values $f(s^*) = P(i, C_{s^*})$ and $f(s_t) = P(i, C_{s_t})$ by running the corresponding quantum circuits many times, and building a probability distribution out of the observed outputs. Decide whether to accept s^* or not according to the SA algorithm.
4. Continue iterating until you find a sentence s that passes a high threshold τ along the objective function i.e. $f(s) > \tau$. This indicates that the sentence is with high probability about the topic i as required.

3.3 Application to Music Composition

Much like how a sentence is composed of words placed side by side, a musical composition can be seen as a sequence of music snippets placed next

²We could even consider the infinite search space of all possible sentences. However, current limitations in quantum hardware mean that solving this more complicated version of the problem is out of scope for the foreseeable future.

to each other. Each snippet itself is in turn composed of musical notes, similarly to how a word is composed of letters belonging to an alphabet.

This similarity was recently exploited in (Miranda et al., 2021) and used to define a musical version of the DisCoCat framework. The authors then used a CFG to generate a data-set of 100 musical compositions for piano. The generated pieces were annotated manually and placed into one of two classes: rhythmic or melodic. This allowed them to train a quantum classifier that distinguishes rhythmic and melodic musical compositions using the techniques of section 2.3.

By replacing the sentence classifier mentioned in section 3.2 with the musical classifier described above, we can adapt our SA based algorithm for the task of generating musical compositions. In the future we will make musical compositions created using this technique available on our project Github repository.

4 Experiments

We now define and attempt to solve two simple sentence generation tasks using the algorithm from the previous section. Our source code is available at <https://bit.ly/QuantumNLG>. To the best of our knowledge, the only other algorithm that can solve these tasks using a quantum computer is what we shall refer to as the Random Generation and Testing (RGT) method of Miranda et al. (2021). In fact, this algorithm was initially proposed for music composition rather than sentence generation, but it can straightforwardly be adapted to perform the latter task as well. It works by randomly putting words from a vocabulary next to each other, and evaluating the resulting sentence against the objective function we defined in section 3, until a satisfactory sentence is found. We will implement sentence generation using RGT and compare its performance with our SA based algorithm.

We do not perform any comparison with state-of-the-art classical methods for solving NLG tasks since it is clear that such methods could easily outperform our proof-of-concept algorithm.

4.1 Food vs IT

For our first task, we use the food vs IT data-set created in Lorenz et al. (2021). This dataset consists of 130 sentences generated using a simple Context-Free Grammar (CFG). Each sentence is manually labelled as being about one of two possible topics,

Food or IT. In Lorenz et al. (2021) a quantum classifier is trained using this dataset according to the techniques discussed in section 2.3. With the help of this classifier, we can implement analyse the SA and RGT based sentence generation algorithms on the Food vs IT dataset.

4.1.1 Simulation results

Before performing experiments on real quantum hardware we first run our algorithms on a ‘classical simulator’. As the name suggests, this is a classical device that simulates the behaviour of a real quantum computer. Of course, it is prohibitively expensive to simulate large quantum systems (otherwise there would be no point in building quantum devices). Fortunately, the quantum circuits we are dealing with in this paper are all very small, and can thus be simulated efficiently. All simulations in this section were performed on a 2019 MacBook Air with 16 GB of memory and a 1.6 GHz Dual-Core Intel Core i5 processor.

As is standard within NLG literature (Sai et al., 2020) we evaluate the quality of free-form generated sentences using the following two criteria:

1. Correctness: Does the generated sentence have the correct topic?
2. Fluency: Is the generated sentence grammatically and semantically correct?

Table 1 shows the result of using a classical simulator to generate 30 sentences about food. The correctness and fluency of each of these sentences has been determined according to the human judgement of the authors. For instance, the sentence “man debugs software” was judged as being fluent but incorrect while the sentence “tasty person prepares dinner” was judged as being correct but not fluent.

	RGT	SA
Fluent and Correct	23	22
Fluent and Not Correct	0	0
Not Fluent and Correct	4	4
Not Fluent and Not Correct	3	4
Avg No. of guesses	7.56	7.46

Table 1: Results of using a classical simulator to generate 30 sentences about food (Number of guesses refers to the number of candidate sentences evaluated against the objective function by each algorithm).

We can see that both the RGT and SA algorithms have performed similarly in terms of the quality of

the produced sentences. This is to be expected given that the acceptance condition for a candidate sentence ($f(s) > \tau$) is the same in both cases. We can also see that the average number of sentences guessed before a valid solution is found is almost the same for both algorithms. This is somewhat surprising, given the more rudimentary nature of RGT compared to SA. We believe the reason for this is the small search space associated with this generation task, as well as the fact that many sentences in this space are actually about food. Thus, RGT has a high likelihood of finding a good sentence in only a few guesses. On the other hand, a poor initial guess in the SA algorithm can be very detrimental in this case, since the algorithm might get stuck in a sub-optimal neighbourhood for a few steps. As we shall see in the news headline generation task, this advantage of RGT quickly disappears when dealing with more complicated search spaces.

4.1.2 Quantum hardware results

We now repeat the experiment above on a real quantum computer, namely IBMs 16 qubit `ibmq_guadalupe` device. When performing experiments on real quantum hardware, it is important to remember that measuring the final state of a quantum circuit will cause this state to collapse to one of the basis states. This means that the only way we can calculate the probabilities $P(i, C_s)$ needed in step 3 of our generation algorithm is to run and measure the circuit C_s repeatedly and create a probability distribution of the observed outcomes. The total number of times a quantum circuit is run in this way is referred to as the number of *shots*. In our case, we ran each circuit for 100000 shots. In the ideal case, results from real quantum hardware will be equivalent to those of simulations. However, imperfections in current prototype devices will lead to sub-optimal performance. The results can therefore be used to benchmark the capacity of current devices for applications of this type.

Table 2 shows the results of using both the RGT and SA algorithms on real quantum hardware in order to generate 10 sentences about food. Interestingly, these results are very similar to the ones obtained using classical simulators in the previous section. This suggests that our algorithms are potentially robust against the inherent noisiness and imperfections of the current generation of quantum computers. We will aim to test this hypothesis further with more extensive future experimentation.

	RGT	SA
Fluent and Correct	7	7
Fluent and Not Correct	0	0
Not Fluent and Correct	2	1
Not Fluent and Not Correct	1	2
Avg No. of guesses	8.4	8.5

Table 2: Results of using the 16 qubit `ibmq_guadalupe` quantum computer to generate 10 sentences about food.

4.2 News Headlines

As we have seen both the SA and RGT based sentence generation algorithms performed fairly well on the Food vs IT dataset. In this section, we will test the behaviour of these algorithms on a more challenging dataset consisting of 105 news headlines. Similarly to (Lorenz et al., 2021), we generated this dataset by using a CFG. The sentences were then manually annotated as belonging to one of four possible news headline topics, *entertainment*, *politics*, *sports*, or *technology*. Compared to the Food vs IT dataset this dataset contains more sentence topics, has a larger vocabulary, and has more complicated CFG production rules. When it comes to sentence generation, this means that there is a much larger search space to consider and that there are fewer acceptable sentences in this search space, making the task significantly more challenging.

Table 3 shows the results of using SA and RGT to generate 30 sentences about politics. As expected for this more complex dataset, the average number of guesses before finding a viable candidate is much less when using SA rather than RGT³.

	RGT	SA
Timeouts	8	0
Fluent and Correct	1	11
Fluent and Not Correct	4	1
Not Fluent and Correct	3	5
Not Fluent and Not Correct	14	13
Avg No. of guesses	201.1	40.4

Table 3: Results of using a classical simulator to generate 30 sentences about politics (Timeout refers to runs of the algorithm that failed to find a suitable sentence after 500 guesses)

³Note that we treat timeouts as 500 guesses for the purposes of averaging.

5 Related and Future Work

We have presented a proof-of-concept algorithm showing how a simple NLG task can be performed on current quantum devices. The algorithm also works for generating musical compositions. Two pieces of related work are worth pointing out:

- In Abbaszade et al. (2021) a hybrid quantum-classical algorithm based on DisCoCat is described for sentence translation, a task which has a language generation component to it. Even though the authors do not provide an implementation, this algorithm is well-suited for experimentation on current quantum hardware, as it relies on Quantum Long Short Term Memory (Q-LSTM) (Chen et al., 2020), a quantum machine learning model that is particularly well-suited for near term devices, due to having a modest requirement on qubit counts and circuit depth.
- Arya et al. (2022) formulates the task of music composition as a Quadratic Unconstrained Binary Optimisation (QUBO) problem. QUBO problems are particularly well-suited for being solved using adiabatic quantum computation (AQC) (Farhi et al., 2000). This is an alternative to the circuit-based model we learnt about in section 2⁴. (Arya et al., 2022) then proceeds to solve this QUBO problem using D-Wave quantum computers and generate musical compositions. In future work, it would be interesting to compare this approach to the RGT and SA algorithms we have discussed here.

We conclude with some thoughts on future research directions.

Clearly, all the works above are limited by the small size of today’s quantum computers. However, several companies have announced plans for building significantly more powerful quantum devices in the next few years (see e.g. qua, 2020). These devices will undoubtedly be capable of solving more sophisticated NLG tasks than the ones presented here. Whether or not this will eventually lead to quantum algorithms that outperform today’s state-of-the-art classical NLG techniques is a fascinating open question that could have dramatic consequences for the field as a whole. We hope

⁴Although both models are equivalent in terms of computational power (Aharonov et al., 2008).

that this work serves as sufficient inspiration for the rest of the community to join us in tackling this question.

A further limitation of our techniques is the fact that DisCoCat, while well-suited for modelling the meaning of sentences, is not capable of modelling the meaning of larger pieces of text. This is problematic when it comes to performing more sophisticated NLG tasks e.g. text summarization, given that these tasks often require the production or manipulation of long passages of text. To alleviate this issue, we could use a recently proposed generalisation of DisCoCat, referred to as the Distributional Compositional Circuit-based (DisCoCirc) model (Coecke, 2021). Inspired by how DisCoCat uses the grammatical relationship between words to encode the meaning of a sentence, DisCoCirc uses the relationship between sentences to encode the meaning of an entire passage of text. A potential avenue for future work is thus to use DisCoCirc and create a pipeline similar to what we have seen in sections 2.3 and 3 for solving document-level rather than sentence-level NLG tasks.

References

2020. IBM’s roadmap for scaling quantum technology. <https://research.ibm.com/blog/ibm-quantum-roadmap>. Accessed: 2022-03-15.
- Scott Aaronson and Lijie Chen. 2016. Complexity-theoretic foundations of quantum supremacy experiments. *arXiv preprint arXiv:1612.05903*.
- Mina Abbaszade, Vahid Salari, Seyed Shahin Mousavi, Mariam Zomorodi, and Xujuan Zhou. 2021. Application of quantum natural language processing for language translation. *IEEE Access*, 9:130434–130448.
- Samson Abramsky and Bob Coecke. 2004. A categorical semantics of quantum protocols. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science, 2004.*, pages 415–425. IEEE.
- Dorit Aharonov, Wim Van Dam, Julia Kempe, Zeph Landau, Seth Lloyd, and Oded Regev. 2008. Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM review*, 50(4):755–787.
- Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. 2019. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510.
- Ashish Arya, Ludmila Botelho, Fabiola Cañete, Dhruvi Kapadia, and Özlem Salehi. 2022. Music composition using quantum annealing. *arXiv preprint arXiv:2201.10557*.
- Zvika Brakerski, Venkata Koppula, Umesh Vazirani, and Thomas Vidick. 2020. Simpler proofs of quantumness. *arXiv preprint arXiv:2005.04826*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Samuel Yen-Chi Chen, Shinjae Yoo, and Yao-Lung L Fang. 2020. Quantum long short-term memory. *arXiv preprint arXiv:2009.01783*.
- Bob Coecke. 2021. The mathematics of text structure. In *Joachim Lambek: The Interplay of Mathematics, Logic, and Linguistics*, pages 181–217. Springer.
- Bob Coecke, Giovanni de Felice, Konstantinos Meichanetzidis, and Alexis Toumi. 2020. Foundations for near-term quantum natural language processing.
- Bob Coecke and Aleks Kissinger. 2018. Picturing quantum processes. In *International Conference on Theory and Application of Diagrams*, pages 28–31. Springer.
- Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. 2010. Mathematical foundations for a composi-

- tional distributional model of meaning. *ArXiv*, abs/1003.4394.
- Ronald De Wolf. 2019. Quantum computing: Lecture notes. *arXiv preprint arXiv:1907.09415*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.
- Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. 2014. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*.
- Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. 2000. Quantum computation by adiabatic evolution. *arXiv preprint quant-ph/0001106*.
- V. Granville, M. Krivanek, and J.-P. Rasson. 1994. [Simulated annealing: a proof of convergence](#). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):652–656.
- Lov K. Grover. 1996. [A fast quantum mechanical algorithm for database search](#). In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, STOC '96*, page 212–219, New York, NY, USA. Association for Computing Machinery.
- Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. 2019. [Supervised learning with quantum-enhanced feature spaces](#). *Nature*, 567(7747):209–212.
- Chris Heunen and Jamie Vicary. 2019. *Categories for Quantum Theory: an introduction*. Oxford University Press.
- Dimitri Kartsaklis, Ian Fan, Richie Yeung, Anna Pearson, Robin Lorenz, Alexis Toumi, Giovanni de Felice, Konstantinos Meichanetzidis, Stephen Clark, and Bob Coecke. 2021. [lambeq: An efficient high-level python library for quantum nlp](#).
- S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. 1983. Optimization by simulated annealing. *Science*, 220(4598):671–680.
- Joachim Lambek. 2008. *From Word to Sentence: a computational algebraic approach to grammar*. Polimetrica sas.
- Xianggen Liu, Lili Mou, Fandong Meng, Hao Zhou, Jie Zhou, and Sen Song. 2020. [Unsupervised paraphrasing by simulated annealing](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 302–312, Online. Association for Computational Linguistics.
- Robin Lorenz, Anna Pearson, Konstantinos Meichanetzidis, Dimitri Kartsaklis, and Bob Coecke. 2021. [Qnlp in practice: Running compositional models of meaning on a quantum computer](#).
- Konstantinos Meichanetzidis, Alexis Toumi, Giovanni de Felice, and Bob Coecke. 2020. [Grammar-aware question-answering on quantum computers](#).
- Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. 2019. Cgmh: Constrained sentence generation by metropolis-hastings sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6834–6842.
- Eduardo Reck Miranda, Richie Yeung, Anna Pearson, Konstantinos Meichanetzidis, and Bob Coecke. 2021. [A quantum natural language processing approach to musical intelligence](#).
- Michael A Nielsen and Isaac Chuang. 2002. Quantum computation and quantum information.
- Edwin Pednault, John A. Gunnels, Giacomo Nannicini, Lior Horesh, and Robert Wisnieff. 2019. Leveraging Secondary Storage to Simulate Deep 54-qubit Sycamore Circuits. *ArXiv:1910.09534*.
- Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'brien. 2014. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5(1):1–7.
- John Preskill. 2018. [Quantum Computing in the NISQ era and beyond](#). *Quantum*, 2:79.
- Ananya B. Sai, Akash Kumar Mohankumar, and Mitesh M. Khapra. 2020. [A survey of evaluation metrics used for nlg systems](#).
- Peter W. Shor. 1997. [Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer](#). *SIAM Journal on Computing*, 26(5):1484–1509.
- Seyon Sivarajah, Silas Dilkes, Alexander Cowtan, Will Simmons, Alec Edgington, and Ross Duncan. 2020. [tket\): a retargetable compiler for nisq devices](#). *Quantum Science and Technology*, 6(1):014003.
- J.C. Spall. 1998. [Implementation of the simultaneous perturbation algorithm for stochastic optimization](#). *IEEE Transactions on Aerospace and Electronic Systems*, 34(3):817–823.
- James R. Wootton. 2020a. [Procedural generation using quantum computation](#). *FDG '20*, New York, NY, USA. Association for Computing Machinery.
- James R. Wootton. 2020b. [A quantum procedure for map generation](#). In *2020 IEEE Conference on Games (CoG)*, pages 73–80.
- Richie Yeung and Dimitri Kartsaklis. 2021. [A ccg-based version of the discocat framework](#).

Daiwei Zhu, Gregory D Kahanamoku-Meyer, Laura Lewis, Crystal Noel, Or Katz, Bahaa Harraz, Qingfeng Wang, Andrew Risinger, Lei Feng, Debopriyo Biswas, et al. 2021. Interactive protocols for classically-verifiable quantum advantage. *arXiv preprint arXiv:2112.05156*.

Towards Evaluation of Multi-party Dialogue Systems

Khyati Mahajan

UNC Charlotte

kmahaja2@uncc.edu

Sashank Santhanam

UNC Charlotte

ssantha1@uncc.edu

Samira Shaikh

UNC Charlotte

sshaikh2@uncc.edu

Abstract

Recent research in the field of conversational AI has emphasized the need for standardization of the metrics used in evaluation. In this work, we focus on evaluation methods used for multi-party dialogue systems. We present an expanded taxonomy focusing on multi-party dialogue based on the need for evaluation dimensions that address challenges associated with the presence of multiple participants. We also survey the evaluation metrics utilized in current multi-party dialogue research, and present our findings with regards to inconsistencies within existing work. Furthermore, we discuss the subsequent need to have more consistent evaluation methodologies and benchmarks. We motivate how consistency will contribute towards a better understanding of progress in the field of multi-party dialogue systems.

1 Introduction

There has been much discussion lately in the field of Natural Language Generation (NLG) focusing the need for evaluation benchmarks and standards, as evidenced by the prolific literature focusing on the issues surrounding human evaluation (Howcroft et al., 2020; Belz et al., 2020; Clark et al., 2021; Hämäläinen and Alnajjar, 2021; van der Lee et al., 2021), as well as recently proposed benchmarks (Gehrmann et al., 2021; Khashabi et al., 2021; Liu et al., 2021; Mille et al., 2021). These are important and necessary debates - however, work has focused mainly on two-party dialogue systems. Multi-party dialogue (MPD) systems, which aim to model conversations between groups (>2 participants) have received less attention, especially in the area of evaluation. Additionally, while there is existing work towards modeling MPD, evaluation strategies are not consistent across existing literature, making it harder to place the progress of the field. In the context of multi-party dialogue (MPD), we discuss both automatic and human evaluation metrics used

for evaluating the three main sub-tasks described in detail in Section 2.

Thus, in this paper, we foreground the challenges faced by the presence of multiple participants in a conversation, and how this property affects the evaluation of systems which aim to model group conversations. We present an expansion to the integrated taxonomy (Table 1) proposed by Higashinaka et al. (2021). We use (Higashinaka et al., 2021) as a baseline owing to their extensive study of data-driven and theory-driven error analysis, and the empirical validation of the proposed integrated taxonomy drawn from both these error analysis paradigms (Higashinaka et al., 2015, 2019). However, we find that the integrated taxonomy does not account specifically for the challenges faced by MPD modeling systems, and thus we propose an expansion specifically keeping these challenges in mind.

We then draw attention to specific shortcomings of evaluation metrics utilized in existing work, such as the lack of consistent reporting within similar evaluation metrics (such as $Recall_n@k$), and the lack of public availability of the proposed methodologies, making it harder to place the progress of the field even if an evaluation benchmark is proposed. Thus, there is a severe gap towards a consistent evaluation framework in Multi-Party Dialogue (MPD) which needs to be addressed. Our main contributions include:

1. We propose an expanded taxonomy focusing on the specific challenges introduced by multi-party dialogue, or group conversations (such as the need to maintain speaker-specific context and recognize the proper addressees), and provide examples for each newly introduced category.
2. We synthesize evaluation measures currently used in MPD research, and relate them to the expanded taxonomy introduced.

To study evaluation metrics in existing work, we surveyed over 338 research papers in the field

of MPD (Github link¹). We obtained the initial pool based on a keyword search for variations of “multi-party dialogue”, with 258 papers focused on work in English, and most of them published at *CL, LREC, and related conferences. The papers included in this article include only those which (a) focus on the English language, (b) include *multiple speakers in the majority of conversations*, and (c) which focus on text-based approaches (thus excluding research which uses multi-modal cues towards the aforementioned sub-tasks). This paper does NOT focus on multilingual corpora, or approaches which solely focus on concepts such as speech recognition or synthesis. We also limit discussion to research published within the past decade for a more relevant understanding of the current progress in MPD modeling, and aim to build upon limited prior work in MPD evaluation, which we discuss further in Section 3.3. With this filtering, we find a total of 15 papers whose aim is one or more of the sub-tasks of Speaker Identification, Addressee Recognition and Response Selection/Generation.

We first present an expanded taxonomy with error reporting drawn from the challenges presented in MPD (Traum, 2003) and (Branigan, 2006), adding categories specifically relevant and important towards MPD evaluation to the taxonomy presented by (Higashinaka et al., 2021). Next, we observe the evaluation metrics utilized in existing work in Section 4, whose error reporting strategies we relate to the proposed expanded taxonomy (Table 1) and note the lack of evaluation for important categories.

2 Overview: Challenges in MPD Evaluation

Evaluation for MPD has often focused on specific sub-tasks that are integral to the working of any conversational system participating in a group conversation. A lot of existing research focuses either on one or more of the sub-tasks: 1) *Speaker Identification* which concerns with how an MPD chatbot is able to track the speakers for each utterance as well as predict who the next speaker could be, 2) *Response Selection* which concerns with the selecting the correct next utterance from a set of choices or *Response Generation* which concerns with generating the next utterance from scratch given the context of the conversation, and 3) *Addressee Recog-*

nition which concerns with being able to find the addressee(s) for the next utterance. All Speaker Identification, Response Selection and Addressee Recognition can be framed as classification tasks (evaluation would need to check whether the correct participant(s) were chosen from the group), whereas Response Generation requires evaluation metrics similar to response generation for two-party dialogue. Recently, systems trained towards jointly modeling one or more of the above tasks have been proposed, however as mentioned before the evaluation strategies lack consistency, and require further thought. While evaluating the classification could provide important indicators of the performance of the dialogue model itself, robust evaluation is needed to understand how well the system would perform in a real life setting. Some leading questions which venture into this challenge faced by MPD systems include:

1. Is the system able to maintain long-term context from all participants in the group? Is the selected/generated response relevant to the prompt and the context of the MPD participants while being grounded in the ongoing conversation? (Pointing to the need for managing speaker information)
2. Is the system able to respond to every participant’s prompt, whether implicitly or explicitly mentioned? Conversely, is it able to learn to not respond (yet remember for context) to the relevant utterances? (Pointing to the need for managing addressee information)
3. Does the system contribute towards making the conversation successful? This success could be attributed to either making the conversation easier for the group by providing information when needed, measuring the interactivity introduced by the presence of MPD dialogue systems, and helping the group achieve the objective which led to the conversation. (Pointing to the need for evaluating appropriate timing and thread management abilities)

Keeping these challenges in mind, we present an expansion of error reporting categories which would be the first step towards accounting for the performance of a system which operated in the multi-party conversation. We briefly summarize the error reporting taxonomy for dialogue agents presented by (Higashinaka et al., 2021), and then discuss how the expansion accounts for errors specific to multi-party dialogue in Section 3.

¹<https://github.com/khyatimahajan/mpd-references>

3 Expanded Taxonomy of Errors for Multi-Party Dialogue

Recently, Higashinaka et al. (2021) introduced an integrated taxonomy of errors in chat-oriented dialogue systems (Table 1). Their work focuses on responses given by a chatbot (conversing with one user) which could cause a breakdown in the conversation (Higashinaka et al., 2015). They empirically validate the resulting integrated taxonomy by asking the same annotators who annotated breakdowns to rate the breakdown for each error category (Higashinaka et al., 2019). While the resulting taxonomy is quite exhaustive, we find that it does not account for challenges specific to MPD, such as the need to know whether the user is able to attribute utterances to each participant correctly. Thus, we expand the taxonomy presented by Higashinaka et al. (2021), focusing specifically on how the presence of multiple participants affects the possible errors which occur in a group conversation.

We elaborate on each error from a MPD point of view, providing examples demonstrating the need for further research. We draw from perspectives presented by Traum (2003) and Branigan (2006), relating the challenges presented for realizing the differences between two-party and multi-party dialogue evaluation. Specifically, we expand on Response-level errors (I18 and I19) which are affected by the speaker and addressee(s), and add a new dimension with Participant-level errors (I20, I21, and I22), which showcase errors from a participant point of view. We include all these, italicized and highlighted, in Table 1, and include details for each error with examples in this section.

3.1 Response-level Errors

This subsection focuses on response level errors, which apply to the semantic meaning of the complex information contained in responses in MPD.

3.1.1 Violation of Content

We maintain the definition presented in Higashinaka et al. (2021), and thus violation of content errors indicate that even though the surface form of the utterance may be appropriate, it could lead to confusion during the conversation.

(I18) Forgot speaker: The utterances made by a specific user are often ignored. This relates specifically to the challenge of Speaker Identification (Traum, 2003), and is an extremely important property for maintaining context in MPD, since it could

create confusion for the system downstream if the utterance is referred to again and the user feels ignored. In the example below, the System (S) forgets the utterance made by User 1 (U1) in the beginning of the conversation. Failure to remember the correct speaker for an utterance could lead to critical downstream errors.

- (1) U1: We need to consider factors A and B for making a decision in case X.
- U2: Factor C would also be interesting and important to consider along with A and B.
- S: U2 mentions factor C will be important to take into consideration for case X.

(I19) Forgot addressee(s): The system forgets to mention the correct addressee(s), relating to the Addressee Recognition challenge (Traum, 2003), and specifically forgets one or more addressees it should have mentioned. If the system was prompted by multiple speakers on a similar topic, but the system responded only to some, this counts as an error since it could make forgotten participants feel alienated from the conversation. In the example below the System (S) forgets to address User 2 (U2), although it should have included both U1 and U2.

- (2) U1: We need to consider factors A and B for making a decision in case X.
- U2: Factor C would also be interesting and important to consider along with A and B.
- S: Thanks for bringing factors A, B and C up for case X, U1.

3.2 Participant-level Errors

We introduce a new broad category of errors towards MPD evaluation called Participant-level errors. The categories of errors introduced in this section stem from the inherently entangled nature of responses in MPD - a response contains not only the content and context of the utterance but also information for who the speaker(s) and addressee(s) of the utterance are. We elaborate further in this subsection.

3.2.1 Violation of Form

The violation of form at the participant level indicates that the system involves the wrong partici-

	Violation of Form	Violation of Content
Utterance	(I1) Uninterpretable (I2) Grammatical error	(I3) Semantic error (I4) Wrong information
Response	(I5) Ignore question (I6) Ignore request (I7) Ignore proposal (I8) Ignore greeting	(I9) Ignore expectation (I18) <i>Forgot speaker</i> (I19) <i>Forgot addressee(s)</i>
Context	(I10) Unclear intention (I11) Topic transition error (I12) Lack of information	(I13) Self-contradiction (I14) Contradiction (I15) Repetition
Society	(I16) Lack of sociality	(I17) Lack of common sense
Participant	(I20) <i>Wrong speaker</i> (I21) <i>Wrong addressee(s)</i>	(I22) <i>Wrong thread response</i> (I23) <i>Inappropriately timed initiative</i>

Table 1: Integrated taxonomy for errors in chat-oriented dialogue systems by Higashinaka et al. (2021). We expand the taxonomy to include errors specific to MPD - extensions are italicized and highlighted in grey. The numbering is assigned serially and used in text to refer to discussions surrounding the specific error.

part(s) in the current state, leading to confusion.

(I20) Wrong speaker: The system credits the wrong speaker for an utterance, also relating to Speaker Identification (Traum, 2003), but specifically mistaking one user for another. This can lead to very important mistakes, since different participants could express differing views which could be contrasting in nature. In the example, the System (S) credits the utterance to U1 instead of U2, which diminishes U2’s contributions.

- (3) U1: We need to consider factors A and B for making a decision in case X.
 U2: Factor C would also be interesting and important to consider along with A and B.
 S: U1 mentions factors C will be taken into consideration for case X.

(I21) Wrong addressee(s): The system mentions the wrong addressee when responding to a prompt, leading to miscommunication. Relating to the challenge of Addressee Recognition (Traum, 2003) as well, mentioning the wrong addressee could lead to a breakdown since this means the system was unable to maintain contextual information which it should have in the conversation. This could lead to a breakdown especially if the addressee who is mentioned does not wish to be mentioned/take part in the current conversation. In the example the System (S) mentions the wrong addressee U1 instead of U2.

- (4) U1: We need to consider factors A and B for making a decision in case X.
 U2: Factor C would also be interesting and important to consider along with A and B.
 S: Interesting insight on factor C, U1.

3.2.2 Violation of Content

A violation of content means that the system makes an error which might seem appropriate in the conversation, but is incorrectly placed, therefore leading to confusion.

(I22) Wrong thread response: MPD can have communication ongoing in multiple threads within the same conversation (Thread/Conversation Management in Traum (2003)). If the system talks about the wrong topic when participating in a different thread, this could lead to confusion and interrupt the desired flow of conversation. In the example below there exist two threads of conversation: one whose topic is sports (U1, U2, U3) and the other whose topic is movies (U4, U5). There are sub-groups of users within the conversation who are participating in different threads, and the System (S) makes an error by mentioning a topic in the wrong thread and sub-group.

- (5) U1: This football season has been going great!
- U2: I agree, for most teams anyway. Which one is your favorite?
- U3: I prefer soccer instead. Anyone here a soccer fan?
- U4: I don't really pay much attention to sports. My main hobby is movies!
- U5: Yeah, and Knives Out was a great one!
- S: I agree U5! The Rams are doing so well this year!

(I23) Inappropriately timed initiative: MPD systems need to figure out when to take the floor in a conversation without causing an abrupt change in the conversation. Secondly, while they could be prompted to speak, it is also important to take the lead to get a conversation started since participants could be yielding the floor to other participants. This relates specifically to the challenge of Initiative Management (Traum, 2003), since the system needs to learn when to take initiative and introduce new topics without which the conversation might come to a halt. In the example the conversation flow is smoothly going on for fiction (U1, U2, and U3), but the System (S) interrupts with a contrasting topic.

- (6) U1: I love documentaries and it has been great seeing so many come out in recent years.
- U2: They do seem informative. I'm particularly interested in performative documentaries, they seem more personal.
- U3: I also enjoy performative documentaries, like Supersize Me. Have you watched it U2?
- S: Does anyone here like fiction?

3.3 Discussion

In recent research, we observe the prevalence of the aforementioned errors within MPD research. We notice how the need to account for multiple participants affects the response selection/generation pipeline for systems modeling MPD, and thus discuss error reporting in existing research in the section to highlight our observations. Since there is limited existing research in the field of MPD response selection/generation, we reserve experimen-

tal validation of the expanded taxonomy for future work. However, one research paper of particular interest to this discussion is Traum et al. (2004, 2006). They are the first to propose evaluations for interactions between virtual multi-party systems and users: 1) User Satisfaction via rated survey questions (accounting for Response-level errors I5-I9, I18, & I19, Society-level errors I16 & I17, and Participant-level errors I20-I23); 2) Intended Task Completion via predefined task success and inter-rater reliability (accounting for I4 and I12); 3) Recognition Rate via classification F-score (accounting for I19 and I21); and 4) Response Appropriateness via a custom defined scale (accounting for Context-level errors I10-I15 and I22-I23). This paper presents a great first step in evaluations for MPD systems which interact in the real world, and we hope to draw from such studies for future work (Section 5).

4 Inconsistency of Evaluation Metrics in Existing Research

Papers focusing on specific tasks within MPD have been observed to employ mostly automatic evaluation measures, with very few including human evaluations. Repeated observations within mainly two-party NLG evaluation have shown that automatic and human evaluations do not correlate well (Belz and Reiter, 2006; Reiter and Belz, 2009; Novikova et al., 2017; Santhanam and Shaikh, 2019; Santhanam et al., 2020), leading to arguments about automatic evaluations being unsuitable for assessing linguistic properties (Scott and Moore, 2007). Owing to these, van der Lee et al. (2021) survey the field and present arguments towards how the inclusion of human evaluations gives a more complete picture of the performance of systems whose main purpose is to participate in human conversations. With research in MPD severely lacking this reporting, it is difficult to place the success of systems which have been proposed to perform well in real-world scenarios. Moreover, owing to the complex nature of group conversations, this lack of reporting exacerbates the effect towards understanding the progress of MPD. Thus, this section illustrates research focusing on the core task of MPD modeling, drawing attention to the evaluation strategies followed by them. We provide a brief synthesis on currently formalized tasks, and relate the errors from the expanded taxonomy (Table 1).

4.1 Evaluation Metrics in Sub-tasks

We organize this section by including sub-task focused discussions to get a clearer idea of the evaluations reported for each sub-task, and how these relate to the expanded taxonomy of errors. We start with the joint formalized task introduced by Ouchi and Tsuboi (2016) - Addressee Recognition and Response Selection, Section 4.1.1 - which is the one of the most consistent research area with regards to error reporting. We then focus specifically on Response Selection in Section 4.1.2, then moving to Response Generation in Section 4.1.3, and lastly Speaker Identification in Section 4.1.4. Lastly, we wrap up by discussing the overall takeaways in Section 4.2.

4.1.1 Addressee Recognition and Response Selection

Ouchi and Tsuboi (2016) first formalized the task of Addressee and Response Selection (ARS) as a joint task, with the input consisting of the (responding agent, context, candidate responses) and the output consisting of the (addressee, response). They evaluate accuracy of their Dual Encoder based RNN model (called Dynamic RNN) over addressee selection (ADR) limited to the addressee of the last utterance, and response selection (RES), as well as a mix of both with addressee-response pair selection (ADR-RES). Zhang et al. (2018b) utilize the same framework for their evaluation, improving their model by including speaker embeddings, called SI-RNN. Le et al. (2019) focus on identifying addressees within the same task, but for all utterances, also reporting accuracy (with n-grams, $n=5, 10, \text{ and } 15$) and $Precision@1$. They additionally involve limited human evaluations, comparing the consistency between human and model outputs, along with significance tests. Gu et al. (2021) introduce MPC-BERT, introducing pre-trained models and fine-tuning for downstream tasks within MPD systems. They follow the same evaluation strategy established by Zhang et al. (2018b).

Thus most papers in this line of research focus on measuring errors towards I18, I19, I20, and I21, with some including human evaluations for a subjective understanding of the success of their models.

4.1.2 Response Selection

Wang et al. (2020) and Gu et al. (2020) focus on response selection as a classification task, with the former proposing Topic-BERT and the latter

proposing SA-BERT, two very similar frameworks. The main difference between the approaches is that Topic-BERT build topic-sentence pairs as input, while SA-BERT instead build speaker embeddings as input - both utilize the basic embeddings for BERT pre-training (segment, position, and token embeddings). Both report recall as defined by the response selection task proposed in DSTC-8² (Kim et al., 2019) sub-tasks 1 and 2, using $Recall_n@k$ for reporting recall for matching n available candidates to k best-matched responses (the official leaderboard utilizes MRR and $Recall@10$ with $n = 100$). However, there is still no overlap in the evaluation results for response selection on DSTC-8 reported by both papers, with Wang et al. (2020) reporting $Recall@1, Recall@5, Recall@10$ and MRR (assuming all these are reported for $n = 100$ - only mentioned in Section 4.1 of the paper) which details the pre-training for Topic-BERT; and Gu et al. (2020) reporting only $Recall_2@1, Recall_{10}@1, Recall_{10}@2, \text{ and } Recall_{10}@5$, although they do mention $Recall_{100}@1$ once in Section 1. Both papers do however mention $Recall_{10}@1, Recall_{10}@2, \text{ and } Recall_{10}@5$ for the Ubuntu V1 corpus, which does allow partial comparison for results. Additionally, Wang et al. (2020) also report BLEU (Papineni et al., 2002) and $Precision@n$ ($n=1, 2, 3, 4$) scores for incorrectly selected responses, checking the relevance of the Topic-BERT retrieved results.

Jia et al. (2020) also tackle response selection, with more focus on dialogue dependency to organize the conversation into contextually aware threads, proposing the Thread-Encoder model (built with Transformer based BERT-base, same as Wang et al. (2020) and Gu et al. (2020)). They utilize similar data (Ubuntu V2 and DSTC-8), and report evaluations for response selection, reporting $hits@k$ (similar to $Recall@k$ as per the paper and ParlAI³ metrics, $k = 1, 2, 5$), and MRR for Ubuntu V2 and $hits@k$ (similar to $Recall@k, k = 1, 5, 10, 50$) and MRR for DSTC-8 (with $n=100$).

Since most papers working on response selection essentially work on a classification task, naturally the reporting is limited to classification metrics. However, even research conducted around the same time, over the same task, reports different metrics with only partial overlaps which could be used to partially compare performance. However, we do

²<https://github.com/dstc8-track2/NOESIS-II/>

³https://parl.ai/docs/tutorial_metrics.html

not consider this evaluation to count towards any of the expanded taxonomy since none of the classification metrics specifically look for performance consciously in any of the dimensions included in the taxonomy - they just measure whether the system was able to choose the next utterance given the previous utterances and a possible list of the right next utterance. Breaking down the evaluation into components presented in the taxonomy, i.e. measuring success keeping in mind the speaker, addressee, and content & context of the selected utterance would help understand the performance in a more robust manner - like Wang et al. (2020) report BLEU for the incorrect responses.

4.1.3 Response Generation

Zhang et al. (2018a), Liu et al. (2019) and Hu et al. (2019) tackle response generation, taking in previous utterances as input and the next utterance as output (Liu et al. (2019) also specifically include the responding speaker and target addressee in the inputs and outputs). Zhang et al. (2018a) report the BLEU- n (n based on n -grams, $n = 1, 2, 3, 4$) and METEOR (Banerjee and Lavie, 2005) scores (mentioning that the evaluation could be supplemented); Liu et al. (2019) report BLEU, ROUGE (Lin, 2004), noun mentions, and length of generated response, along with limited human evaluations for fluency, consistency, and informativeness; and Hu et al. (2019) report BLEU- n ($n = 1, 2, 3, 4$), METEOR, ROUGE-L (L for longest common subsequence), along with human evaluations for fluency, grammaticality, and rationality. Qiu et al. (2020) focus on the dialogue thread structures which are utilized in Hu et al. (2019), utilizing structured attention with Variational RNN, reporting the same automatic metrics BLEU- n ($n = 1, 2, 3, 4$), METEOR, ROUGE-L (L for longest common subsequence). They also find that they are able to perform speaker identification and addressee recognition without specifically training towards these tasks.

Yang et al. (2019) tackle response generation along with speaker identification, proposing LSTMs to build an encoder, a contextual RNN, a speaker encoder, and a decoder, called Multi-role Interposition Dialogue System (MIDS). They report accuracy for speaker identification; and perplexity and loss for response generation.

Even with the majority of papers reporting the basic automated evaluation metrics most common for generation (BLEU, METEOR, ROUGE

(van der Lee et al., 2021)), these are not always reported. Moreover, Liu et al. (2016) also show that the aforementioned metrics show either weak or no correlation with human judgements. Human evaluations are also limited, although they do cover some of the most reported metrics (fluency, consistency, informativeness, grammaticality, rationality (van der Lee et al., 2021)). Most research thus cover major aspects of the expanded taxonomy, namely Utterance-level I1-I4, Context-level I10-I15, and Society-level I17. Some papers also report speaker identification and addressee recognition, accounting for I18, I19, and Participant-level I20-I23 with thread management.

4.1.4 Speaker Identification

Ma et al. (2017) and de Bayser et al. (2019) focus on speaker identification, with the former using RNN and CNN to identify speakers with a sitcom dataset, and the latter using MLE, SVM, CNN, and LSTM architectures to model sitcom, finch and multibotwoz datasets. While both utilize a variety of features (such as surrounding utterance concatenation, agent and content information) with the models to improve predictions, Ma et al. (2017) report accuracy and F1 (+ F1 towards each participant and a confusion matrix to better analyze wrong predictions), and de Bayser et al. (2019) report accuracy. They extend their work in de Bayser et al. (2020) by integrating MLE, CNN, and FSA-based architectures, for multibotwoz, reporting accuracy.

Classification for speaker identification does help response selection and generation, counting towards errors I18 and I20 from the expanded taxonomy. However, it would be helpful to include more classification metrics (like Ma et al. (2017) who report the confusion matrix) to allow for more robust evaluations.

4.2 Discussion

It is imperative to observe the various kinds of evaluation metrics which have been used to evaluate different tasks within MPD. Most metrics reported are not consistent across the main task they focus on, sometimes even when they report performance on a shared task such as DSTC-8 (Kim et al., 2019). It is important to note that these inconsistencies lead to confusion when it comes to looking for the current state-of-the-art systems, as well as for making important performance comparisons such as significance testing. Additionally, we find that there is a 50-50% (8:7) division of the code in the

papers being publicly available (if we include broken links, the unavailability goes up, but we count these as attempts to provide reproducible methods). This means that even with re-evaluation given a benchmark, there is a possibility that comparison across existing research will not be able to provide a full picture of the progress in each sub-task.

All these issues draw attention to the need for more shared tasks and robust benchmarks which report errors in a manner fitting the proposed taxonomy. We postulate that this would allow better comparisons across tasks, and overall performance towards building systems able to participate in MPD - although we reserve the evaluation of our proposed extensions to the taxonomy itself for future work. We aim to follow methods similar to the ones described by (Higashinaka et al., 2019) to maintain the standards they set up for validation of error analysis.

5 Conclusion and Future Work

We have presented an expansion - which focuses specifically on errors important in multi-party dialogue - to the integrated taxonomy of errors proposed by Higashinaka et al. (2021). We include examples for each newly introduced error in Section 3, and relate the errors to the challenges detailed by Traum (2003). We then present inconsistencies in the evaluation strategies reported in existing research (Section 4), organized by the sub-tasks they focus on. We observe the difficulty in comparisons across the proposed methods owing to inconsistencies in error analysis. We also relate the reported errors to the expanded taxonomy, drawing parallels for an overall comparison.

We observe how the challenges introduced by the presence of multiple participants affect the need for more robust evaluations (Section 3.3) which are capable of reporting how well the approach performs, and find that (Traum et al., 2004, 2006) provide a great discussion surrounding these errors, albeit more focused on interactions between virtual systems and users. We also find that even with defined tasks, inconsistencies could arise in reporting errors (Section 4.2), leading to confusion when placing the progress of research in MPD.

We note that while our presented taxonomy is relevant to the errors reported in current literature, there is a need to evaluate their effectiveness empirically, which is the main limitation for this paper and proposed future work. Another big limita-

tion of this work which is also a part of proposed future work is the formalization of the proposed expanded errors specific to MPD from this paper (Table 1), and the validation of the formalization towards a proposed benchmark. The first shared task DSTC-8 (Kim et al., 2019) focused on the response selection sub-task, however there is the need for future shared tasks which account for all three sub-tasks (speaker identification, response selection/generation and addressee recognition), and related sub-tasks (such as disentanglement, thread management, and coreference resolution).

References

- Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.
- Anja Belz, Simon Mille, and David M Howcroft. 2020. Disentangling the properties of human evaluation methods: A classification system to support comparability, meta-evaluation and reproducibility testing. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 183–194.
- Anja Belz and Ehud Reiter. 2006. Comparing automatic and human evaluation of nlg systems. In *11th conference of the european chapter of the association for computational linguistics*, pages 313–320.
- Holly P. Branigan. 2006. Perspectives on multi-party dialogue. *Research on Language and Computation*, 4:153–177.
- Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A Smith. 2021. All that’s ‘human’ is not gold: Evaluating human evaluation of generated text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7282–7296.
- Maíra Gatti de Bayser, P. Cavalin, C. Pinhanez, and Bianca Zadrozny. 2019. Learning multi-party turn-taking models from dialogue logs. *ArXiv*, abs/1907.02090.
- Maira Gatti de Bayser, Melina Alberio Guerra, Paulo Cavalin, and Claudio Pinhanez. 2020. A hybrid solution to learn turn-taking in multi-party service-based chat groups. *Interactions*, 10(4):2.
- Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Aremu Anuoluwapo, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna Clinciu, Dipanjan Das, Kaustubh D

- Dhole, et al. 2021. The gem benchmark: Natural language generation, its evaluation and metrics. *arXiv preprint arXiv:2102.01672*.
- Jia-Chen Gu, Tianda Li, Quan Liu, Zhen-Hua Ling, Zhiming Su, Si Wei, and Xiaodan Zhu. 2020. Speaker-aware bert for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2041–2044.
- Jia-Chen Gu, Chongyang Tao, Zhenhua Ling, Can Xu, Xiubo Geng, and Daxin Jiang. 2021. Mpc-bert: A pre-trained language model for multi-party conversation understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3682–3692.
- Mika Härmäläinen and Khalid Alnajjar. 2021. The great misalignment problem in human evaluation of nlp methods. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 69–74.
- Ryuichiro Higashinaka, Masahiro Araki, Hiroshi Tsukahara, and Masahiro Mizukami. 2019. Improving taxonomy of errors in chat-oriented dialogue systems. In *9th International Workshop on Spoken Dialogue System Technology*, pages 331–343. Springer.
- Ryuichiro Higashinaka, Masahiro Araki, Hiroshi Tsukahara, and Masahiro Mizukami. 2021. Integrated taxonomy of errors in chat-oriented dialogue systems. In *SIGDIAL*.
- Ryuichiro Higashinaka, Kotaro Funakoshi, Masahiro Araki, Hiroshi Tsukahara, Yuka Kobayashi, and Masahiro Mizukami. 2015. Towards taxonomy of errors in chat-oriented dialogue systems. In *Proceedings of the 16th annual meeting of the special interest group on discourse and dialogue*, pages 87–95.
- David M Howcroft, Anja Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A Hasan, Saad Mahamood, Simon Mille, Emiel van Miltenburg, Sashank Santhanam, and Verena Rieser. 2020. Twenty years of confusion in human evaluation: Nlg needs evaluation sheets and standardised definitions. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 169–182.
- Wenpeng Hu, Zhangming Chan, Bing Liu, Dongyan Zhao, Jinwen Ma, and Rui Yan. 2019. Gsn: A graph-structured network for multi-party dialogues. In *IJCAI*.
- Qi Jia, Yizhu Liu, Siyu Ren, Kenny Zhu, and Haifeng Tang. 2020. Multi-turn response selection using dialogue dependency relations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1911–1920.
- Daniel Khashabi, Gabriel Stanovsky, Jonathan Bragg, Nicholas Lourie, Jungo Kasai, Yejin Choi, Noah A Smith, and Daniel S Weld. 2021. Genie: A leaderboard for human-in-the-loop evaluation of text generation. *arXiv preprint arXiv:2101.06561*.
- Seokhwan Kim, Michel Galley, Chulaka Gunasekara, Sungjin Lee, Adam Atkinson, Baolin Peng, Hannes Schulz, Jianfeng Gao, Jinchao Li, Mahmoud Adada, et al. 2019. The eighth dialog system technology challenge. *arXiv preprint arXiv:1911.06394*.
- Ran Le, Wenpeng Hu, Mingyue Shang, Zhenjun You, Lidong Bing, Dongyan Zhao, and R. Yan. 2019. Who is speaking to whom? learning to identify utterance addressee in multi-party conversations. In *EMNLP/IJCNLP*.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Cao Liu, Kang Liu, Shizhu He, Zaiqing Nie, and Jun Zhao. 2019. Incorporating interlocutor-aware context into response generation on multi-party chatbots. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 718–727.
- Chia-Wei Liu, Ryan Lowe, Iulian Vlad Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132.
- Pengfei Liu, Jinlan Fu, Yang Xiao, Weizhe Yuan, Shuaichen Chang, Junqi Dai, Yixin Liu, Zihuiwen Ye, and Graham Neubig. 2021. Explainaboard: An explainable leaderboard for nlp. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 280–289.
- Kaixin Ma, Catherine Xiao, and Jinho D. Choi. 2017. Text-based speaker identification on multiparty dialogues using multi-document convolutional neural networks. In *ACL*.
- Simon Mille, Kaustubh Dhole, Saad Mahamood, Laura Perez-Beltrachini, Varun Gangal, Mihir Kale, Emiel van Miltenburg, and Sebastian Gehrmann. 2021. Automatic construction of evaluation suites for natural language generation datasets. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.
- Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for nlg. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252.

- Hiroki Ouchi and Yuta Tsuboi. 2016. Addressee and response selection for multi-party conversation. In *EMNLP*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Liang Qiu, Yizhou Zhao, Weiyan Shi, Yuan Liang, Feng Shi, Tao Yuan, Zhou Yu, and Song-chun Zhu. 2020. Structured attention for unsupervised dialogue structure induction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1889–1899.
- Ehud Reiter and Anja Belz. 2009. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, 35(4):529–558.
- Sashank Santhanam, Alireza Karduni, and Samira Shaikh. 2020. Studying the effects of cognitive biases in evaluation of conversational agents. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13.
- Sashank Santhanam and Samira Shaikh. 2019. [Towards best experiment design for evaluating dialogue system output](#). In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 88–94, Tokyo, Japan. Association for Computational Linguistics.
- Donia Scott and Johanna Moore. 2007. An nlg evaluation competition? eight reasons to be cautious. In *Proceedings of the Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation*, pages 22–23.
- D. Traum. 2003. Issues in multiparty dialogues. In *Workshop on Agent Communication Languages*.
- David R Traum, Susan Robinson, and Jens Stephan. 2004. Evaluation of multi-party virtual reality dialogue interaction. In *LREC*.
- David R Traum, Susan Robinson, and Jens Stephan. 2006. Evaluation of multi-party reality dialogue interaction. Technical report, UNIVERSITY OF SOUTHERN CALIFORNIA MARINA DEL REY CA INST FOR CREATIVE
- Chris van der Lee, Albert Gatt, Emiel van Miltenburg, and Emiel Kraemer. 2021. Human evaluation of automatically generated text: Current trends and best practice guidelines. *Computer Speech & Language*, 67:101151.
- Weishi Wang, Steven CH Hoi, and Shafiq Joty. 2020. Response selection for multi-party conversations with dynamic topic tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6581–6591.
- Qichuan Yang, Z. He, Zhiqiang Zhan, Jianyu Zhao, Y. Zhang, and C. Hu. 2019. Mids: End-to-end personalized response generation in untrimmed multi-role dialogue*. *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.
- Haisong Zhang, Zhangming Chan, Yan Song, Dongyan Zhao, and R. Yan. 2018a. When less is more: Using less context information to generate better utterances in group conversations. In *NLPCC*.
- Rui Zhang, H. Lee, L. Polymenakos, and Dragomir R. Radev. 2018b. Addressee and response selection in multi-party conversations with speaker interaction rns. In *AAAI*.

Are Current Decoding Strategies Capable of Facing the Challenges of Visual Dialogue?

Amit Kumar Chaudhary

CIMeC, University of Trento
amitkumar.chaudhar@unitn.it

Alex J. Lucassen

CIMeC, University of Trento
alex.lucassen@unitn.it

Ioanna Tsani

CIMeC, University of Trento
ioanna.tsani@unitn.it

Alberto Testoni

DISI, University of Trento
alberto.testoni@unitn.it

Abstract

Decoding strategies play a crucial role in natural language generation systems. They are usually designed and evaluated in open-ended text-only tasks, and it is not clear how different strategies handle the numerous challenges that goal-oriented multimodal systems face (such as grounding and informativeness). To answer this question, we compare a wide variety of different decoding strategies and hyper-parameter configurations in a Visual Dialogue referential game. Although none of them successfully balance lexical richness, accuracy in the task, and visual grounding, our in-depth analysis allows us to highlight the strengths and weaknesses of each decoding strategy. We believe our findings and suggestions may serve as a starting point for designing more effective decoding algorithms that handle the challenges of Visual Dialogue tasks.

1 Introduction

The last few years have witnessed remarkable progress in developing efficient generative language models. The choice of the decoding strategy plays a crucial role in the quality of the output (see [Zarrieß et al. \(2021\)](#) for an exhaustive overview). It should be noted that decoding strategies are usually designed for and evaluated in text-only settings. The most-used decoding strategies can be grouped into two main classes. On the one hand, decoding strategies that aim to generate text that maximizes likelihood (like greedy and beam search) are shown to generate generic, repetitive, and *degenerate* output. [Zhang et al. \(2021\)](#) refer to this phenomenon as *the likelihood trap*, and provide evidence that these strategies lead to sub-optimal sequences. On the other hand, stochastic strategies like pure sampling, top-k sampling, and nucleus sampling ([Holtzman et al., 2020](#)) increase the variability of generated texts by taking random samples from the model. However, this comes at the cost of generating words

that are not semantically appropriate for the context in which they appear. Recently, [Meister et al. \(2022\)](#) used an information-theoretic framework to propose a new decoding algorithm (typical decoding), which samples tokens with an information content close to their conditional entropy. Typical decoding shows promising results in human evaluation experiments but, given its recent release, it is not clear yet how general this approach is.

Multimodal vision & language systems have recently received a lot of attention from the research community, but a thorough analysis of different decoding strategies in these systems has not been carried out. Thus, the question arises of whether the above-mentioned decoding strategies can handle the challenges of multimodal systems. i.e., generate text that not only takes into account lexical variability, but also grounding in the visual modality. Moreover, in goal-oriented tasks, the informativeness of the generated text plays a crucial role as well. To address these research questions, in this paper we take a referential visual dialogue task, *GuessWhat?! (De Vries et al., 2017)*, where two players (a Questioner and an Oracle) interact so that the Questioner identifies the secret object assigned to the Oracle among the ones appearing in an image (see [Figure 1](#) for an example). Apart from well-known issues, such as repetitions in the output, this task poses specific challenges for evaluating decoding techniques compared to previous work. On the one hand, the generated output has to be coherent with the visual input upon which the conversation takes place. As highlighted by [Rohrbach et al. \(2018\)](#); [Testoni and Bernardi \(2021b\)](#), multimodal generative models often generate *hallucinated* entities, i.e., tokens that refer to entities that do not appear in the image upon which the conversation takes place. On the other hand, the questions must be informative, i.e., they must help the Questioner to incrementally identify the target object.

We show that the choice of the decoding strat-



<u>Questioner</u>	<u>Oracle</u>
Is it a vase?	Yes
Is it partially visible?	No
Is it in the left corner?	No
Is it the turquoise and purple one?	Yes

Figure 1: Example of a GuessWhat game from De Vries et al. (2017)

egy and its hyper-parameter configuration heavily affects the quality of the generated output. Our results highlight the specific strengths and weaknesses of decoding strategies that aim at generating sequences with the highest probability vs. strategies that randomly sample words. We find that none of the decoding strategies currently available is able to balance task accuracy and linguistic quality of the output. However, we also show which strategies perform better at important challenges, such as incremental dialogue history, human evaluation, hallucination rate, and lexical diversity. We believe our work may serve as a starting point for designing decoding strategies that take into account all the challenges involved in Visual Dialogue tasks.

2 Task & Dataset

GuessWhat?! (De Vries et al., 2017) is a simple object identification game in English where two participants see a real-world image from MSCOCO (Lin et al., 2014) containing multiple objects. One player (the Oracle) is secretly assigned one object in the image (the target) and the other player (the Questioner) has to guess it by asking a series of binary yes-no questions to the Oracle. The task is considered to be successful if the Questioner identifies the target. The dataset for this task was collected from human players via Amazon Mechanical Turk. The authors collected 150K dialogues with an average of 5.3 binary questions per dialogue. Figure 1 shows an example of a GuessWhat game from the dataset.

3 Model and Decoding Strategies

We use the model and pre-trained checkpoints of the Questioner agent made available by Testoni and Bernardi (2021c) for the GuessWhat?! task. This model is based on the GDSE architecture (Shekhar et al., 2019). It uses a ResNet-152 network (He et al., 2016) to encode the images and an LSTM network to encode the dialogue history. A multi-modal shared representation is generated and then used to train both the question generator (which generates a follow-up question given the dialogue history) and the Guesser module (which selects the target object among a list of candidates at the end of the dialogue) in a joint multi-task learning fashion. Testoni and Bernardi (2021c) added an internal Oracle module to the GDSE architecture, which guides a cognitively-inspired beam search re-ranking strategy (*Confirm-it*) at inference time: this strategy promotes the generation of questions that aim at confirming the model’s intermediate conjectures about the target. In our work, at inference time the Questioner agent always interacts with the baseline Oracle agent proposed in De Vries et al. (2017).

We analyse the effect of a large number of decoding strategies as well as hyper-parameter configuration for each strategy: as highlighted by Zhang et al. (2021), it is crucial to evaluate different hyper-parameter configurations when comparing multiple decoding strategies. Among the ones that maximize the likelihood of the sequence, we consider plain **beam search** (with a beam size of 3) and **greedy search**. We also consider **Confirm-it**, the cognitively-inspired beam search re-ranking strategy proposed in Testoni and Bernardi (2021c) for promoting the generation of questions that aim at confirming the model’s intermediate conjectures about the target. This strategy re-ranks the set of candidate questions from beam search and selects the one that helps the most in confirming the model’s hypothesis about the target. As for stochastic strategies, we analyse **pure sampling**, **top-k sampling** (with different k values), and **nucleus sampling** (with different p values), a strategy proposed in Holtzman et al. (2020) which selects the highest probability tokens whose cumulative probability mass exceeds a given threshold p . We also consider **typical decoding** (with different τ values), a recently proposed strategy (Meister et al., 2022) based on an information-theoretic framework. We refer to the respective papers for additional details

on decoding strategies. We let the model generate 5 questions¹ at test time and average the results over five random seeds.

4 Metrics

We are interested in evaluating different decoding strategies against a set of metrics that reflect the complexity of the different skills required to successfully solve multimodal referential games.

Linguistic Quality: We compute the percentage of games with at least one repeated question, the overall number of unique words used by the model and, in line with the observations in Testoni and Bernardi (2021a), the number of *rare words* generated by the model, defined as those words that appear fewer than 20 times in the training set.

Visual Grounding: To quantify the rate of object hallucination in the generated dialogues, we compute the CHAIR metric (Rohrbach et al., 2018; Testoni and Bernardi, 2021b). This metric, originally proposed for image captioning, detects hallucination by checking each object mentioned in a generated image caption against the ground-truth MSCOCO objects for that image. The metric consists of two distinct variants: CHAIR-i, or per-instance variant (number of hallucinated objects divided by the total number of objects mentioned in each dialogue), and CHAIR-s, or per-sentence variant (number of dialogues with at least one hallucination divided by the total number of dialogues).²

Informativeness: To study the informativeness of the generated questions, we report the raw accuracy of the model in guessing the target object after each dialogue turn and at the end of the dialogue. A game is considered successful if the model identifies the target object assigned to the Oracle. Similarly, we also report the accuracy of human annotators when guessing the target by reading machine-generated dialogues.

5 Results

5.1 Quantitative Results

Table 1 shows the performance of different decoding strategies against accuracy and dialogue quality, as described by the metrics in Section 4.³ Confirm-

¹Except for the accuracy per turn metric in Section 5.3, where the dialogues consisted of 10 questions.

²Testoni and Bernardi (2021b) first adapted the CHAIR metric for Visual Dialogue. However, the authors did not investigate the effect of different decoding strategies.

³Here we only report the best-performing configuration for each decoding strategy (see SM for all configurations).

it is by far the best decoding strategy in terms of accuracy and hallucination rate. However, it uses a restricted vocabulary compared to other strategies. A similar issue is observed for greedy and beam search. We find nucleus sampling (with a p -value of 0.3, much lower than the one used by the authors in Holtzman et al. (2020)) to effectively increase the lexical variety compared to beam search, without damaging accuracy and hallucination rate. Typical decoding, top-k and pure sampling, instead, clearly decrease repetitions and increase the vocabulary richness by generating tokens that are not related to the source input, as indicated by the high hallucination rate. It thus looks like there exists a trade-off between informativeness / visual grounding and linguistic quality.

5.2 Effect of Hyper-Parameter Choice

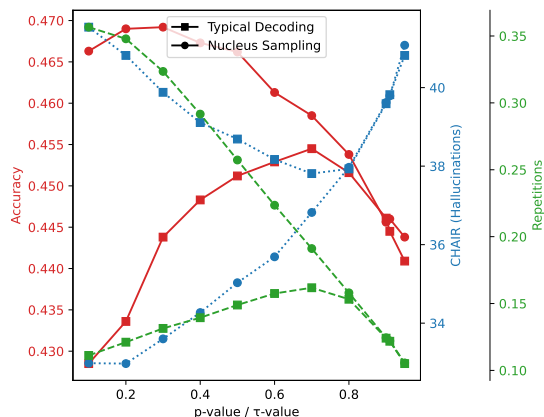


Figure 2: Different hyper-parameter values and their effect on the accuracy, hallucinations, and repetitions in typical decoding and nucleus sampling.

We study the effect of hyper-parameter configurations in stochastic strategies. Specifically, we try various p -values for nucleus sampling and τ -values for typical decoding.⁴ As shown in Figure 2, both typical and nucleus sampling peak in accuracy with the parameter configurations that also lead to the most repetitions and fewest hallucinations. Conversely, both strategies show the lowest accuracy with the highest hallucination rate. These results confirm the detrimental effect of hallucinations on the performance of the model. It is interesting to note the robustness of typical decoding in generating few repetitions regardless of the τ value. In line with the findings in Zhang et al. (2021), this analysis confirms the importance of hyper-parameter

⁴Results for top-k are in SM.

	Accuracy (%) \uparrow	CHAIR-i \downarrow	CHAIR-s \downarrow	% games with repetitions \downarrow	Vocabulary Size \uparrow	Rare Words \uparrow
Confirm-it	51.39	15.09	28.48	30.33	858	34
Beam Search (beam size = 3)	47.05	18.33	31.08	38.49	731	27
Nucleus Sampling ($p = 0.3$)	46.92	17.96	33.60	32.35	1016	78
Greedy Search	46.58	17.75	32.97	35.63	834	46
Typical Decoding ($\tau = 0.7$)	45.45	21.84	37.81	16.18	1703	247
Top-k Sampling ($k = 5$)	45.10	22.84	37.71	14.93	1462	171
Pure Sampling	43.13	26.55	43.23	8.32	2609	793

Table 1: Comparison between decoding strategies and their best-performing (in terms of accuracy) hyper-parameters. The decoding strategies are sorted by accuracy.

configurations and the peculiar trade-off between informativeness, repetitions, and visual grounding: so far it has not been possible to find a single configuration that optimizes all three at the same time.

5.3 Per-turn Accuracy

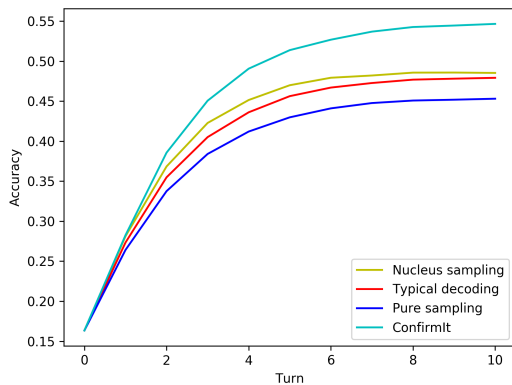


Figure 3: The accuracy per dialogue turn for four different decoding strategies for dialogues of length 10.

One crucial ability in GuessWhat?! is asking informative questions that incrementally help in identifying the target: for this reason, we check the accuracy of the model after each new question is asked. Figure 3 shows accuracy per dialogue turn for a set of representative strategies: Nucleus sampling ($p=0.3$), Typical Decoding ($\tau=0.7$), Confirm-it, and pure sampling. To get a broader picture, we let the model generate 10 questions in this setting. Confirm-it stands out by showing the largest incremental increase of accuracy throughout the dialogue, indicating that it generates more effective follow-up questions. Pure sampling, on the other hand, seems to suffer from the very beginning of the dialogue and its accuracy stabilizes soon. It is worth noting that the accuracy of typical decoding gets closer to that of nucleus sampling towards

	Human Accuracy (%) \uparrow
Confirm-it	72.5
Typical Sampling ($\tau = 0.7$)	68.0
Nucleus Sampling ($p = 0.3$)	67.5
Pure Sampling	59.5

Table 2: Human Guess Accuracy based on dialogue generated from different decoding strategies.

the end of the dialogue, with the latter leveling off sooner. We conjecture that Confirm-it outperforms other techniques because it takes into account the probability of the Guesser at inference time, so it is guided to generate questions that change these probabilities and thus avoid generic questions.

5.4 Human Evaluation

We asked 8 human annotators to guess the target object in a sample of GuessWhat?! games when reading dialogues generated by our model with different decoding strategies. Each participant annotated 100 games (25 per strategy) and the decoding strategy was not revealed during the annotation. As shown in Table 2, humans reach the highest accuracy when reading dialogues generated by Confirm-it, followed by typical decoding and nucleus sampling, while pure sampling falls behind. These results, which do not mirror the accuracy result in Table 1, allow us to disentangle the weaknesses of the Guesser (i.e., the classification module that predicts the target) from the actual informativeness of the dialogues. Compared to the model, human annotators seem to better exploit the lexical richness of typical decoding and nucleus sampling. We refer to the SM for additional information about the annotation procedure, in line with the best-practice guidelines in van der Lee et al. (2021).

6 Related Work

In the field of multimodal NLG, [Zarrieß and Schlangen \(2018\)](#) propose trainable decoding for referring expression generation. The authors propose a two-stage optimization set-up where a small network processes the RNN’s hidden state before passing it to the decoder, using BLEU score as a reward for the decoder. We did not analyse this approach in our paper because we focus only on decoding strategies that do not require any change in the architecture or training of the model. We leave for future work an analysis of trainable decoding approaches. Inspired by the findings in [Holtzman et al. \(2020\)](#), [Massarelli et al. \(2020\)](#) propose a hybrid decoding strategy for open-ended text generation which combines the non-repetitive nature of sampling strategies with the consistency of likelihood-based approaches. The authors show that their approach generated less repetitive and more verifiable text. The design of hybrid decoding strategies for multimodal tasks is out of the scope of this paper, but is an interesting subject to pursue in future work.

7 Discussion and Conclusion

Decoding algorithms are a key component of natural language generation systems. They are usually designed for and evaluated in text-only tasks. We believe multimodal (vision & language) and goal-oriented tasks pose unique and under-studied challenges to current decoding strategies. In this paper, we ran an in-depth analysis of several decoding strategies (and their hyper-parameter configurations) for a model playing a referential visual dialogue game. We found that decoding algorithms that lead to the highest accuracy in the task and the lowest hallucination rate, at the same time generate highly repetitive text and use a restricted vocabulary. Our analyses reveal the crucial role of hyper-parameter configuration in stochastic strategies, an issue that poses several questions about the trade-off between lexical variety, hallucination rate, and task accuracy. While nucleus sampling partially balances the above-mentioned issues, human annotators seem to better exploit the richness of the dialogues generated by typical decoding. Finally, our results demonstrate that a beam search re-ranking algorithm (Confirm-it) generates more effective follow-up questions throughout the dialogue turns. We believe that taking into account the model’s intermediate predictions about the ref-

erent, like Confirm-it does, represents a promising direction that should be applied also to stochastic strategies in future work, aiming at preserving their lexical richness while reducing hallucinations.

Our results demonstrate that none of the decoding strategies currently at disposal effectively take into account both task accuracy and dialogue quality at the same time. We also highlight peculiar features of each strategy that may guide future research with the goal of designing decoding strategies that properly confront the crucial challenges of multimodal goal-oriented dialogues.

Acknowledgements

We kindly acknowledge the support of the Department of Information Science and Engineering (DISI) for the arrangement of the GPUs used in our research. We would like to thank Professor Raffaella Bernardi for the suggestions and feedback. The first three authors are enrolled in the Erasmus Mundus European Masters Program in Language and Communication Technologies. The first author is supported by the Erasmus+ Programme. The third author of this scientific paper was supported by the Onassis Foundation - Scholarship ID: F ZR 060-1/2021-2022. We thank the Master’s students of the Grounded Language Processing course at the University of Trento for the feedback provided at an early stage of this project.

References

- Harm De Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron Courville. 2017. [GuessWhat?! Visual object discovery through multi-modal dialogue](#). In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 4466–4475. IEEE Computer Society.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. [Deep residual learning for image recognition](#). In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. [The curious case of neural text degeneration](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. [Microsoft COCO](#):

- common objects in context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer.
- Luca Massarelli, Fabio Petroni, Aleksandra Piktus, Myle Ott, Tim Rocktäschel, Vassilis Plachouras, Fabrizio Silvestri, and Sebastian Riedel. 2020. [How decoding strategies affect the verifiability of generated text](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 223–235, Online. Association for Computational Linguistics.
- Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2022. [Typical decoding for natural language generation](#). *CoRR*, abs/2202.00666.
- Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. [Object hallucination in image captioning](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4035–4045, Brussels, Belgium. Association for Computational Linguistics.
- Ravi Shekhar, Aashish Venkatesh, Tim Baumgärtner, Elia Bruni, Barbara Plank, Raffaella Bernardi, and Raquel Fernández. 2019. [Beyond task success: A closer look at jointly learning to see, ask, and Guess-What](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2578–2587.
- Alberto Testoni and Raffaella Bernardi. 2021a. [The interplay of task success and dialogue quality: An in-depth evaluation in task-oriented visual dialogues](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 2071–2082. Association for Computational Linguistics.
- Alberto Testoni and Raffaella Bernardi. 2021b. [“I’ve seen things you people wouldn’t believe”: Hallucinating entities in GuessWhat?!](#) In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 101–111, Online. Association for Computational Linguistics.
- Alberto Testoni and Raffaella Bernardi. 2021c. [Looking for confirmations: An effective and human-like visual dialogue strategy](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9330–9338, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Chris van der Lee, Albert Gatt, Emiel van Miltenburg, and Emiel Kraemer. 2021. Human evaluation of automatically generated text: Current trends and best practice guidelines. *Computer Speech & Language*, 67:101151.
- Sina Zarrieß and David Schlangen. 2018. [Decoding strategies for neural referring expression generation](#). In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 503–512, Tilburg University, The Netherlands. Association for Computational Linguistics.
- Sina Zarrieß, Henrik Voigt, and Simeon Schüz. 2021. [Decoding methods in neural language generation: A survey](#). *Information*, 12(9):355.
- Hugh Zhang, Daniel Duckworth, Daphne Ippolito, and Arvind Neelakantan. 2021. [Trading off diversity and quality in natural language generation](#). In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 25–33, Online. Association for Computational Linguistics.

A Supplementary Material

A.1 Effect of Hyper-parameters

Figures 4, 5, and 6 illustrate how hyper-parameter choice affects the accuracy, the hallucinations, and the repetitions. Top-k sampling (Figure 4) shows decreased accuracy and repetitions, and increased hallucinations, as the k -value gets higher. The same general pattern can be observed with the gradual increase of the p -value in nucleus sampling (Figure 6). On the other hand, typical decoding accuracy peaks at $\tau = 0.7$ (Figure 5). This is also the point at which the repetitions are at their highest and the hallucinations are at their lowest. Both very high and very low τ -values cause lower accuracy, fewer repetitions, and an increase of hallucinations.

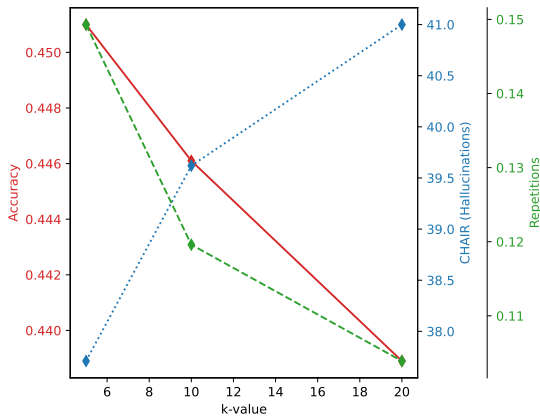


Figure 4: Hyper-parameter choices’ effect on the accuracy, hallucinations, and repetitions in top-k sampling.

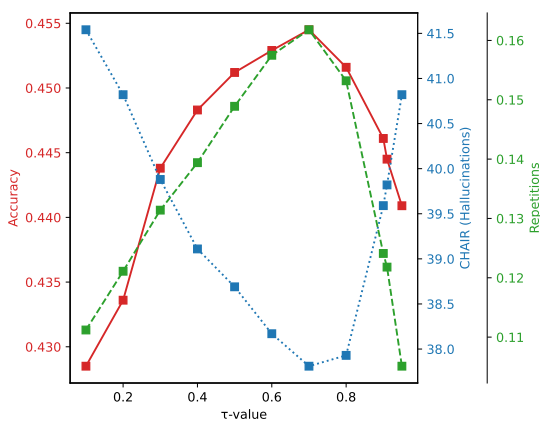


Figure 5: Hyper-parameter choices’ effect on the accuracy, hallucinations, and repetitions in typical decoding.

A.2 Experiments

Table 3 presents our results in detail for all the parameter configurations we considered. We have

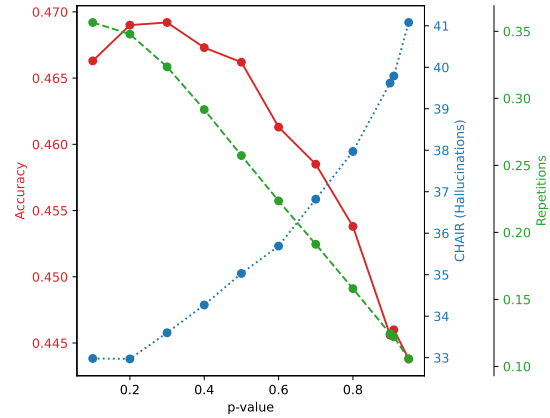


Figure 6: Hyper-parameter choices’ effect on the accuracy, hallucinations, and repetitions in nucleus sampling.

computed accuracy percentage, CHAIR-i, CHAIR-s, percentage of games with repeated questions, vocabulary size and number of rare words for each decoding method and its respective hyper-parameter configurations. These results are sorted by decreasing accuracy. The 3 best results of each metric are in bold.

A.3 Human Annotation Details



Figure 7: Example of the games displayed to the participants for the annotation task. Participants had to select one target object among the list of candidate objects on the right. The machine-generated dialogue is in the red box.

The annotation was done by 8 human annotators on a sample of GuessWhat?! games. They were recruited within our organization on a voluntary basis and they did not receive any payment for the annotation. Written informed consent was obtained from all the participants. Participants were 4 males and 4 females with high educational level and from

	% Accuracy \uparrow		CHAIR-i \downarrow		CHAIR-s \downarrow		% games with repetitions \downarrow		Vocabulary Size \uparrow		Rare Words \uparrow	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
CI	51.39	0.00	15.09	0.00	28.48	0.00	30.33	0.00	858	0.0	34	0.0
BS (beam = 3)	47.05	0.00	18.33	0.00	31.08	0.00	38.49	0.00	731	0.0	27	0.0
NS ($p = 0.3$)	46.92	0.17	17.96	0.09	33.60	0.14	32.35	0.21	1016	8.3	78	5.5
NS ($p = 0.2$)	46.90	0.11	17.65	0.06	34.79	0.12	46.41	0.18	919	7.1	59	4.5
NS ($p = 0.4$)	46.73	0.27	18.38	0.17	34.27	0.28	29.16	0.25	1097	11.9	88	6.4
NS ($p = 0.1$)	46.63	0.02	17.76	0.01	32.98	0.01	35.66	0.06	839	1.6	46	1.9
NS ($p = 0.5$)	46.62	0.17	19.10	0.09	35.03	0.10	25.73	0.26	1192	18.0	103	3.4
GS	46.58	0.00	17.75	0.00	32.97	0.00	35.63	0.00	834	0.0	46	0.0
NS ($p = 0.6$)	46.13	0.38	20.04	0.15	35.69	0.35	22.35	0.26	1303	12.7	126	9.4
NS ($p = 0.7$)	45.85	0.19	21.14	0.11	36.82	0.31	19.11	0.26	1451	9.0	162	9.5
TD ($\tau = 0.7$)	45.45	0.32	21.84	0.15	37.81	0.23	16.18	0.29	1703	13.0	247	12.6
NS ($p = 0.8$)	45.38	0.14	22.20	0.16	37.97	0.30	15.80	0.19	1643	23.0	219	12.9
TD ($\tau = 0.6$)	45.29	0.28	22.08	0.16	38.17	0.30	15.75	0.17	1723	21.9	248	18.3
TD ($\tau = 0.8$)	45.16	0.18	22.21	0.20	37.93	0.29	15.32	0.28	1712	10.8	244	13.6
TD ($\tau = 0.5$)	45.12	0.15	22.60	0.17	38.69	0.36	14.89	0.22	1745	7.3	262	8.7
Top-k ($k = 5$)	45.10	0.27	22.84	0.21	37.71	0.26	14.93	0.10	1462	12.6	171	5.2
TD ($\tau = 0.4$)	44.83	0.17	23.11	0.13	39.11	0.44	13.94	0.24	1755	19.0	265	12.2
TD ($\tau = 0.9$)	44.61	0.16	23.74	0.18	39.59	0.19	12.41	0.25	1919	13.5	334	9.1
Top-k ($k = 10$)	44.61	0.24	24.03	0.29	39.62	0.26	11.96	0.16	1692	13.8	235	10.5
NS ($p = 0.91$)	44.60	0.15	23.92	0.13	39.79	0.23	12.21	0.13	1948	22.3	342	14.0
NS ($p = 0.9$)	44.56	0.23	23.82	0.07	39.62	0.17	12.44	0.10	1912	20.2	332	13.6
TD ($\tau = 0.91$)	44.45	0.27	23.92	0.14	39.82	0.31	12.18	0.19	1945	11.7	345	16.1
TD ($\tau = 0.3$)	44.38	0.31	24.07	0.21	39.88	0.22	13.14	0.23	1791	14.8	278	13.7
NS ($p = 0.95$)	44.38	0.12	24.93	0.15	41.08	0.24	10.56	0.05	2129	11.3	438	11.4
TD ($\tau = 0.95$)	44.09	0.21	24.83	0.24	40.82	0.28	10.51	0.20	2117	18.9	435	17.1
Top-k ($k = 20$)	43.89	0.10	25.12	0.30	41.00	0.39	10.39	0.17	1879	23.1	305	17.5
TD ($\tau = 0.2$)	43.36	0.20	25.19	0.16	40.82	0.34	12.11	0.09	1815	21.5	287	10.7
PS	43.13	0.28	26.55	0.25	43.23	0.36	8.32	0.17	2609	9.3	793	11.4
TD ($\tau = 0.1$)	42.85	0.15	26.25	0.14	41.54	0.09	11.12	0.11	1825	18.4	286	13.5

Table 3: Comparison between decoding strategies and their hyper-parameters (CI = Confirm-it, BS = Beam Search, NS = Nucleus Sampling, GS = Greedy Search, TD = Typical Decoding, Top-k = Top-k Sampling, PS = Pure Sampling).

different ethnic groups. Before the beginning of the annotation task, each annotator was briefed on the GuessWhat?! gameplay and purpose, and was asked to annotate some sample games in order to get familiar with the annotation process. We used the makesense.ai online software for image recognition. Each image had a minimum of 3 and a maximum of 6 candidate objects. The annotators could see both the bounding box and the category for each candidate object in the image. They could also see the full dialogue between the Questioner and the Oracle. The annotators then had to pick the object they believed was the right one, based on the information given by the dialogue. Figure 7 provides an example of the games we asked the participants to annotate. Overall, we extracted 200 images from the GuessWhat?! test set and generated one dialogue per decoding strategy per image. In total, we thus generated 800 dialogues. Each human participant annotated 25 images per decoding strategy. To prevent biases, the participants were not given any information about the decoding strategy that each image represented.

A.4 Comparison of Decoding Strategies

Table 4 shows a comparison between generated dialogues of various decoding strategies (see Figure 8 for the image the dialogues refer to).

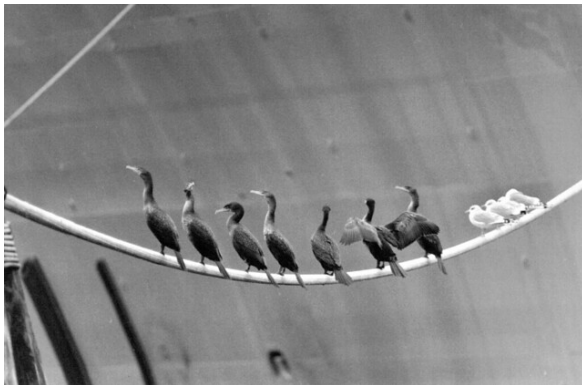


Figure 8: The image for game 147344.

Turn	Speaker	Confirm-it	Nucleus Sampling	Typical Decoding	Pure Sampling
0	Questioner	is it a bird?	is it a bird?	is it a bird?	is it a bird?
	Oracle	yes	yes	yes	yes
1	Questioner	is it on the left side?	is it on the left side?	is it on the air?	is it close?
	Oracle	yes	yes	yes	yes
2	Questioner	is it on the left side of the group?	is it one of the first birds?	is it 1 of the 3 on the right?	is it squatting?
	Oracle	no	yes	no	yes
3	Questioner	is it in the middle?	is it the first one from the left?	is it one of the three birds that are next to each other?	is it one of the three all the way to the right?
	Oracle	yes	no	yes	yes
4	Questioner	is it one of the three on the right?	is it the second one from the left?	is it further left?	is it 1st?
	Oracle	no	no	yes	no

Table 4: Comparison between dialogues generated by various decoding strategies for the image reported in Figure 8.

Analogy Generation by Prompting Large Language Models: A Case Study of InstructGPT

Bhavya Bhavya¹, Jinjun Xiong² and ChengXiang Zhai¹

¹Department of Computer Science, University of Illinois at Urbana-Champaign

¹{bhavya2, czhai}@illinois.edu

²Department of Computer Science and Engineering, University at Buffalo

²jinjun@buffalo.edu

Abstract

We propose a novel application of prompting Pre-trained Language Models (PLMs) to generate analogies and study how to design effective prompts for two task settings: generating a source concept analogous to a given target concept (aka Analogous Concept Generation or ACG), and generating an explanation of the similarity between a given pair of target concept and source concept (aka Analogous Explanation Generation or AEG). We found that it is feasible to prompt InstructGPT to generate meaningful analogies and the best prompts tend to be precise imperative statements especially with a low temperature setting. We also systematically analyzed the sensitivity of the InstructGPT model to prompt design, temperature, and injected spelling errors, and found that the model is particularly sensitive to certain variations (e.g., questions vs. imperative statements). Further, we conducted human evaluation on 1.4k of the generated analogies and found that the quality of generations varies substantially by model size. The largest InstructGPT model can achieve human-level performance at generating meaningful analogies for a given target while there is still room for improvement on the AEG task.¹

1 Introduction

Large Pre-trained Language Models (PLMs) such as BERT (Devlin et al., 2018) and GPT (Brown et al., 2020) have been applied to many tasks of text generation (e.g., summarization, dialogue system) with promising results (Li et al., 2021). However, no existing work has studied how to apply PLMs to generate different kinds of textual analogies, such as conceptual metaphors (e.g., “Life is a journey²”), and instructional analogies (e.g., “A red blood cell is like a truck in that they both transport essential supplies” (Newby et al., 1995)).

¹Our code and datasets are available for public use: <https://github.com/Bhavya/InstructGPT-Analogies>

²https://en.wikipedia.org/wiki/Conceptual_metaphor

Table 1: Selected prompts and InstructGPT-generated analogies for *natural selection*

Prompt (P7):	<i>What is analogous to natural selection?</i>
InstructGPT Output:	The analogous process to natural selection is artificial selection. (9 words)
Prompt (P2):	<i>Explain natural selection using a well-known analogy.</i>
InstructGPT Output:	Imagine that you have a jar of mixed nuts ... If you shake the jar ...the big nuts will fall out first ... analogy is that natural selection is like a sieve that separates the fit from the unfit... (136 words)

Generating analogies has a wide range of applications, such as explaining concepts and scientific innovation, and analogies play a crucial role in human cognition. Analogical matching and reasoning enables humans to understand and learn unfamiliar concepts (aka target concepts) by means of familiar ones (aka source concepts) and to make scientific innovations. Unsurprisingly, analogy modeling and generation has been a long-standing goal of AI (Mitchell, 2021). This is a challenging problem because it often requires computing deep semantic similarities that are beyond the surface-level similarity. For example, the Bohr’s atom model and the solar system are analogous due to their structural and relational similarities (i.e., atoms orbit around the nucleus like planets around the sun).

Much work has been done to compute such analogical similarities between concepts. However, existing approaches mostly rely on structured representations, thus, they can only where such representations already exist. For example, one of the most popular models is Structural Mapping Engine (SME) (Forbus et al., 2017), which aligns *structured representations* of the target and source concepts using predicate logic. Moreover, they cannot *generate* analogies in natural language.

Inspired by the recent success in applying PLMs to many NLP tasks (e.g., (Li et al., 2021)), we propose and study the application of PLMs to analogy generation. We consider two typical application scenarios of analogy generation: 1) Analogous Concept Generation (ACG): given a target concept (e.g., bohr’s model), generate a source concept analogous to the target concept (e.g., solar system), possibly with an explanation of their similarities; 2) Analogy Explanation Generation (AEG): given a target concept and an analogous source concept, generate an explanation of their similarities.

By noting the similarity of the two tasks defined above to other text generation problems, and being inspired by the recent success of using prompted PLMs for text generation, we propose analogy generation by using a PLM with appropriately designed prompts. We adopt the promising emerging paradigm of prompting language models (Liu et al., 2021) that uses textual prompts with unfilled slots and directly leverages the language models to fill those slots and obtain the desired output. For example, Table 1 shows sample prompts and PLM-generated outputs for ACG from our experiments.

Specifically, we study the following main research questions: RQ1) How effective is a modern PLM such as InstructGPT in generating meaningful analogies? RQ2) How sensitive are the generated analogies to prompt design, the temperature hyperparameter, and spelling errors? RQ3) How does the model size impact the quality of generated analogies?

To study these questions, we design several experiments on analogies generated from the InstructGPT (Ouyang et al., 2022) model. First, we manually validate whether InstructGPT can generate meaningful analogies for ten well-known analogies in the science domain. Next, we design and systematically vary prompt variants (e.g., imperative statements vs. questions) and temperature, and investigate the corresponding variations in the generated text by comparing them to a reference dataset of science analogies. Finally, we study the impact of model size on the quality of generated analogies both by automatically comparing against the reference data and using human evaluation.

Our experimental results show that PLMs (specifically, InstructGPT) offer a promising general approach to generating analogies with properly designed prompts. Furthermore, the InstructGPT model is found to be sensitive to the prompt design,

temperature, and spelling errors for this task, particularly to the prompt style (i.e., question vs. imperative statement). Precise imperative statements in low-temperature setting are found to be the best prompts. Finally, the quality of the generated analogies depends heavily on the model size. While the largest model can achieve human-level performance on the ACG task, the smallest model barely generates any meaningful analogies. The AEG task proved to be more challenging based on human evaluation and could be a better test of the analogical reasoning capabilities of PLMs especially for explaining analogies not seen during training.

2 Related Work

2.1 Computational Models of Analogies

There has been a lot of work on computational modeling of analogies (Mitchell, 2021). The SME model (Forbus et al., 2017) is one of the most popular symbolic model that finds the *mapping*, or connections between structured representations of source and target concepts and their attributes. However, such methods cannot generate new analogous source concepts with analogical explanation.

The recent deep learning-based approaches, including using pre-trained language models (Mikolov et al., 2013; Rossiello et al., 2019; Ushio et al., 2021), are able to *generate* analogies to some extent, but are currently limited to simple word-level and proportional analogies, such as (ostrich:bird :: lion:?). In contrast, we aim to generate and explain more complex analogies of concepts, e.g. instructional analogies (Newby et al., 1995).

Another line of work is on finding analogous documents for scientific innovation, such as product descriptions and research papers, based on their semantic similarities (Kittur et al., 2019). In contrast, we operate in a generative task setup.

To the best of our knowledge, none of the existing work has studied the problem of automatically generating complex analogies in natural language. Recently, research on more “generative” analogy-making tasks has been recommended (Mitchell, 2021). Along this direction, we believe that our proposed task is challenging and more practically useful than the existing text-based generative analogical tasks including letter-string (e.g., if “abc” changes “abd”), what does “pqrs” change to?) and word-level analogies.

2.2 Prompting Language Models

Recently, prompts have been either manually created or learned to successfully leverage PLMs for several natural language tasks (Liu et al., 2021). Our work is closest to prompting for lexical and proportional analogy generation (Ushio et al., 2021). But, none of the existing work has performed an in-depth study on prompting PLMs for both generating analogous concepts given a single query concept and explaining the analogical similarities between two query concepts.

3 Problem Formulation

Motivated by the practical applications of this task (e.g., explaining concepts), we study analogy generation in the following settings.

1. Analogous Concept Generation (ACG) or **No Source (NO_SRC)**: Here, only the target concept is provided as the input. The goal is to generate an analogous source concept or scenario, along with some explanation to justify the analogy. For example, “Explain Bohr’s atomic model using an analogy.”

2. Analogy Explanation Generation (AEG) or **With Source (WSRC)**: Here, in addition to the target, the source concept is also a part of input. The goal is to generate an explanation of how the target and source are analogous. For example, “Explain how Bohr’s atomic model is analogous to the solar system.”

Our problem setup is similar to the use of PLMs for text generation (Li et al., 2021), and is most closely related to single-relation analogy generation (e.g., ostrich : bird :: animal : lion) (Ushio et al., 2021), where the input is a pair of query concept (e.g., ostrich : bird), and the task is to choose an analogical pair from a pre-defined list of candidate pairs. But, our proposed task is still different in nature and much more challenging (e.g., requiring more creativity in some cases). First, both of our inputs and outputs are different. For example, in the proposed ACG setup, our input is a single concept (e.g., “bohr’s model”), not a pair of concepts. Our task is to identify another concept (or scenario) that has an equivalence to the query concept based on their deep and non-trivial semantic similarities. No previous work has studied this kind of “single-concept-based” analogy generation with pre-trained language models. Even in the proposed AEG setup where we also use a pair of concepts as input, they are different from the pair used in

the previous work. For example, our input could be a pair (e.g., “bohr’s model” and “solar system”) and the output is an explanation of their analogical relations (e.g., how their structures are similar). Second, we do not have a pre-defined finite list of candidates to choose from, which is a more realistic and interesting setting than previous work from application perspectives, and is also much more challenging for evaluation.

4 Experiment Setup

In this section, we discuss InstructGPT PLM and datasets used in our experiments.

InstructGPT Model: Recently, several PLMs have been developed and trained on massive web data (Devlin et al., 2018; Brown et al., 2020; Raffel et al., 2019). In this study, we probe the aligned GPT-3 models, InstructGPT. These are GPT-3 models that have been optimized to follow instructions better (Ouyang et al., 2022). InstructGPT has four variants depending on the model size (number of parameters), namely Ada (350 M), Babbage (1.3 B), Curie (6.7 B), and Davinci (175 B)³. Unless otherwise mentioned, we use the Davinci model for the experiments as it is expected to have the best performance.

We used the Open AI API⁴ to generate all analogies. Main hyperparameters are described in Section 5.2.2 and rest in the Appendix A

Dataset: As the task of analogy generation, as defined in this paper, has not been previously studied, there is no existing data set available to use directly for evaluation. We thus opted to create new data sets for evaluation. Table 2 shows sample data from these datasets.

Standard Science Analogies (STD): As far as we could find, the closest dataset consisting of conceptual analogies is from (Turney, 2008). It consists of ten standard science analogies. However, these only contain the source and target concepts but not any explanation in natural language.

Science analogies from academic Q&A sites SAQA: We searched for quiz questions that asked to create analogies on academic Q&A sites like Chegg.com, Study.com⁵ by using search queries

³<https://blog.eleuther.ai/gpt3-model-sizes/>

⁴<https://beta.openai.com/docs/api-reference/completions/create>

⁵<https://chegg.com/>, <https://study.com/>. We manually inspected the data and found no personal identifiers or offensive content. We manually compiled the datasets, no scraping was done.

like ‘create an analogy’, ‘analogy to explain’, and manually downloaded the relevant questions and answers. After manually removing irrelevant data, 75 unique question-answer pairs were obtained. Next, we manually extracted the analogies from answers, i.e., target and source concepts, and the explanation of the analogical similarity.

There are total 109 concepts (about high-school science) with 148 English analogies. The average word length of analogies is 62.25 words.

Table 2: Sample analogies from STD and SAQA.

Dataset	Target	Source	Explanation
STD	atom	solar system	-
SAQA	ligase	sewing machine	... Ligase is similar to a sewing machine, as it binds two elements ... (25 words)

5 Experiment Results

In this section, we present our experiment results and examine each of the three research questions introduced earlier.

5.1 Feasibility Analysis

We first examine RQ1 and investigate whether InstructGPT is capable of generating analogies with simple prompts by looking at the results on the smaller STD dataset which contains well-known analogies. Here, we seek standard analogies, so we designed prompts with keywords such as "well-known analogy", "often used to explain", etc. The full list of prompts is in Table 17, Appendix C).

We observed that all the prompts were successful in retrieving natural language analogies to some extent but they differed in several aspects. Table 1 shows sample analogies generated by two of our prompts (P7 and P2, Table 17) for the target concept “natural selection.” In this case, the reference answer in the STD dataset is “artificial selection,” which P7 successfully retrieved, while P2 generated a different but also valid analogy. Such variations indicate both the potential of using different prompts to generate (multiple) different analogies and the model sensitivity to prompt design, which we further investigate in Section 5.2.

To quantify the effectiveness of different prompts, we manually evaluated the source con-

cepts mentioned in the generated analogies (if any). Table 3 shows the number of exact matches of generated source concepts to those in the reference STD dataset, along with the number of “valid” source concepts generated. Valid means a reasonable analogy that is either commonly known (e.g., easily available on the internet ⁶) or contains a meaningful justification. All prompts generated valid analogies in most cases, even if they didn’t exactly match the reference source concept further suggesting the promise of InstructGPT for generating meaningful analogies. Note that the low number of exact matches with the reference dataset is expected to some extent because there are several possible “valid” analogies for a given source concept and so there is a small chance that the model would generate exactly the same analogous concept as in the reference.

Table 3: Number of analogies that match the ground truth or are otherwise meaningful, out of the total ten analogies generated for STD target concepts by the seven prompts (P1-P7).

	P1	P2	P3	P4	P5	P6	P7
# Match	3	3	6	4	3	5	3
# Valid	6	9	9	8	7	10	10

5.2 Robustness analyses

As observed in many other applications of prompted PLMs, the performance of a task tends to be sensitive to the prompts used and the temperature parameter (Lu et al., 2021; Zhao et al., 2021). Moreover, many PLMs are known to be vulnerable to the presence of spelling errors (Pruthi et al., 2019; Ma et al., 2020). Thus, it is important to experiment with variations of both the prompts and the temperature parameter (with frequency_penalty, Section 5.2.2), and spelling errors and study how they impact the generated analogy (RQ2).

For these analyses, we need to compare the model performance in a large number of configurations, which makes human evaluation impossible. Thus, we rely on automatic metrics. Automatic evaluation of natural language generation is known to be challenging (e.g., long-form question answering (Krishna et al., 2021)) and automatic metrics generally have low correlation with human judgment (Callison-Burch et al., 2006; Raffel et al., 2019). Evaluation of analogies is even more chal-

⁶Note that commonly known does not necessarily mean available on the internet. We use it only as a proxy here since there is no good way to determine what is common knowledge.

lenging especially because a target concept could have several valid analogies with seemingly different meanings (e.g., “artificial selection” vs. “sieve” from Section 5.1). Thus, before using existing methods, we designed sanity checks and found that those methods behave as we expect (e.g., analogies have a higher score than non-analogies, see Appendix B). We note that our sanity checks are only the necessary and not the sufficient requirements of a good metric for evaluating analogies as they do not evaluate creativity or reasoning. However, we use them as an approximation only for relative comparison between methods on the same task as they are unlikely to favor any single method.

We use three representative measures of automatic evaluation of generated text: BLEURT (Sellam et al., 2020) (B), METEOR (Lavie and Agarwal, 2007) (M), ROUGE-L (R)⁷ (Lin, 2004)⁸. BLEURT (B) is used as the primary metric for evaluation since it is a recent machine learning-based metric that has been shown to capture semantic similarities between texts (Sellam et al., 2020).

Similar average BLEURT values would indicate that the prompts are equally good (or bad) on a task, but not necessarily in the same way. On the other hand, Kendall’s Tau (Kendall, 1938) indicates how well the ranks of two variables are correlated. This would suggest that those prompts have similar strengths and weaknesses. Thus, we analyze both scores to get a more complete picture of hyperparameter sensitivity.

5.2.1 Analysis of prompts

To study the effectiveness and robustness of different prompts for analogy generation in the unsupervised setting, we manually designed several prompts for all the problem settings. The different prompt variants are all paraphrases that are semantically similar. The main ways they differ are: 1. *Questions vs. Imperative Statements* (e.g., P5 vs. P2, Table 5); 2. *Synonyms* (e.g., P2 vs. P3, Table 5); 3. *Word Ordering* (e.g., P1 vs. P3, Table 4). We only study the zero-shot setting mainly because the choice/number of examples in few-shot could make an impact on the generated analogies and make it harder to interpret our experiment results.

Prompts for the NO_SRC and WSRC settings are in Tables 4,5, respectively. Here, <target>, <src> are target and source concept placeholders.

Our major findings are as follows:

⁷<https://pypi.org/project/rouge-score/>

⁸https://www.nltk.org/api/nltk.translate.meteor_score.html

Table 4: Prompts for NO_SRC

Id	Prompt
P1	Explain <target> using an analogy.
P2	Create an analogy to explain <target>.
P3	Using an analogy, explain <target>.
P4	What analogy is used to explain <target>?
P5	Use an analogy to explain <target>.

Table 5: Prompts for WSRC

Id	Prompt
P1	Explain <target> using an analogy involving <src>.
P2	Explain how <target> is analogous to <src>.
P3	Explain how <target> is like <src>.
P4	Explain how <target> is similar to <src>.
P5	How is <target> analogous to <src>?
P6	How is <target> like <src>?
P7	How is <target> similar to <src>?

Questions and statements are significantly different: The question prompts are P4, Table 4 and P5-P7, Table 5. From Tables 6 and 7, questions have significantly different and lower scores than statements. This could be an artifact of how the InstructGPT models were trained and should be further investigated.

Table 6: Comparison of performances of different prompts and temperatures in NO_SRC. * and † mean statistically significant compared to the best performing setting at p<0.1 and p<0.05 respectively based on a two-tailed t-test.

	B	R	M
P1 _{tl}	0.46	0.187	0.154
P1 _{th}	0.448†	0.181†	0.167
P2 _{tl}	0.451	0.193	0.154
P2 _{th}	0.45*	0.184	0.161
P3 _{tl}	0.462	0.196	0.164
P3 _{th}	0.452	0.188	0.171
P4 _{tl}	0.427†	0.170†	0.126†
P4 _{th}	0.431†	0.179†	0.156
P5 _{tl}	0.451	0.188	0.154
P5 _{th}	0.449*	0.183*	0.163

Impact of synonyms and word order: Prompt performances vary based on synonyms and word order. For example, some synonymous prompt pairs (e.g. P2-P4, P5-P7 in WSRC) are more correlated than others (e.g., P2-P3, P5-P6 in WSRC). This could be because “analogous to” and “similar to” share a word unlike the other synonym “like”. As expected,

prompts with the most different meanings (e.g., P1 in WSRC – involving <src> is not necessarily the same as analogous to <src>) are least correlated with others. However, from Table 7, the average performances of synonymous prompts (e.g., $P2_{tl}$ and $P3_{tl}$, $P2_{tl}$ and $P5_{tl}$) are not significantly different. Overall, this suggests that InstructGPT is more robust to synonyms/word-order than to the prompt style (question/imperative statements) for this task. The overall best-performing prompts (P3 in NO_SRC, P2 in WSRC) contain some form of the word “analogy” rather than its synonyms, confirming that precise and direct prompts are better.

Table 7: Comparison of performances of different prompts and temperatures in WSRC. * and † mean statistically significant at $p < 0.1$ and $p < 0.05$ compared to the best performing setting respectively based on a two-tailed t-test.

	B	R	M
$P1_{tl}$	0.504	0.223	0.187 [†]
$P1_{th}$	0.497 [†]	0.212 [†]	0.199
$P2_{tl}$	0.515	0.217	0.203
$P2_{th}$	0.502*	0.210 [†]	0.208
$P3_{tl}$	0.504	0.229	0.191
$P3_{th}$	0.504	0.216	0.203
$P4_{tl}$	0.506	0.214	0.197
$P4_{th}$	0.497 [†]	0.206 [†]	0.2
$P5_{tl}$	0.499*	0.217	0.18 [†]
$P5_{th}$	0.496 [†]	0.211 [†]	0.191*
$P6_{tl}$	0.500*	0.216	0.176 [†]
$P6_{th}$	0.494 [†]	0.212 [†]	0.183 [†]
$P7_{tl}$	0.497 [†]	0.208 [†]	0.179 [†]
$P7_{th}$	0.492 [†]	0.204 [†]	0.186 [†]

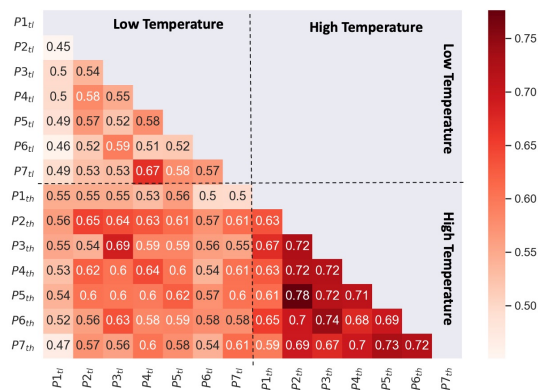


Figure 1: Kendall’s Tau correlation between BLEURT scores of various prompts and temperatures in WSRC

5.2.2 Analysis of temperature

Higher temperature increases the randomness in the generated text and is often suggested for creative tasks (Lucy and Bamman, 2021). Since some analogies require creativity, we are especially interested in studying the impact of this hyperparameter.

We explore two settings. **Low Temperature (tl)**: this is a deterministic setting, where temperature = frequency_penalty = presence_penalty = 0. **High Temperature (th)**: Here temperature is set to 0.85. To avoid repetition of words and topics, we set frequency_penalty = 1.24 and presence_penalty = 1.71. These hyperparameters were selected based on initial qualitative exploration. To account for the randomness, we set best_n = 3, i.e., select the best response out of three generated responses, and generate 5 such best responses. In all experiments, we report the average performance of all 5 responses.

From the results in lower-right corner of Figures 1 and 3 (in Appendix D), we make the following two observations:

Firstly, at high temperature, prompts are generally well-correlated with each other suggesting lesser sensitivity to prompt design at high temperatures. This requires further investigation because we expect higher randomness to generate a variety of different analogies, and thus have lower correlations in general.

Secondly, the overall performances of the high-temperature variants are generally lower than their low-temperature counterparts. To investigate when high temperature could help, we further looked into a case in the WSRC setting where the high-temperature version of the best prompt, ($P2_{th}$), performed much better. The results are shown in Table 8. In this case, unlike $P2_{hl}$, $P2_{tl}$ fails on identifying the target and also generates incorrect facts, (“rubber of your lungs”). This shows some evidence of high temperature prompts working better for more complex and creative analogies, which should be investigated further in the future.

5.2.3 Analysis of synthetic spelling errors

Since obtaining natural spelling errors can be difficult, injecting synthetic errors is a common way to test the robustness of models (e.g., (Jayanthi et al., 2020)). Thus, following previous work (Sakaguchi et al., 2017; Jayanthi et al., 2020), we injected the following four types of character-level errors to the internal characters of the target concept in the prompt: Delete (delete one randomly chosen character), Permute (switch two randomly chosen ad-

Table 8: Analogies generated in the high and low-temperature setting by the prompt: *Explain how breathing mechanism of frogs is analogous to blowing of the balloon.*

Temp.	Generated Analogy
High	... When you blow into a balloon, the air pressure ... The frog’s respiratory system works in a similar way. When you breathe in, you increase the air pressure inside your lungs. This increased air pressure causes the walls ...
Low	When you blow up a balloon, the air inside the balloon is pushing outwards on the rubber of the balloon. This is the same thing that happens when you breathe in: the air inside your lungs is pushing outwards on the rubber of your lungs.

acent characters in the string), Insert (insert one random alphabet at a random position), and Replace (replace one randomly chosen character in the string with a random alphabet). Target concepts with length less than 3 were kept unchanged.

Average BLEURT scores from three different runs for all prompts in the low-temperature setting in NO_SRC are shown in Table 9. Overall, the performance decreases, indicating the sensitivity of language models to spelling errors. Further, Replace generally leads to the biggest performance drop for all prompts ($\sim 3 - 7\%$ relative decrease). The model is generally most robust to Insert, similar to the results reported in previous work on word recognition using neural networks (Sakaguchi et al., 2017).

Table 9: Impact of injecting Delete (D), Permute (P), Insert (I) and Replace (R) errors to the target concept in the prompt compared to the original (O) prompt based on BLEURT scores. * and † mean statistically significant at $p < 0.1$ and $p < 0.05$ respectively based on a two-tailed t-test.

	D	P	I	R	O
P1	0.438†	0.437†	0.436†	0.429†	0.46
P2	0.431†	0.434†	0.442	0.427†	0.451
P3	0.444†	0.445†	0.447*	0.44†	0.462
P4	0.423	0.424	0.428	0.416	0.427
P5	0.438*	0.437*	0.441	0.435†	0.451

5.3 Analysis of model size

Finally, we examine RQ3, i.e., how does the model size impact the quality of the generated analogies. In general, models with more parameters can be expected to perform better. We now study whether the same holds for this task and how much the model size impacts the performance.

Figure 2 shows the BLEURT scores of various models on both the task setups. As expected, the performance increases significantly with model size in both WSRC and NO_SRC, suggesting that

larger models are better at generating analogy-like text for the given targets. Further, the biggest improvement is seen as the number of parameters increases from 0.3B to 1.3B in both settings (19.17% and 15.34% relative improvements, respectively).

Similar to what we observed in the case of the 175B Davinci model, the performance in WSRC is higher than that in NO_SRC for other models too. This confirms that all models have some capacity to incorporate the source provided in the prompt.

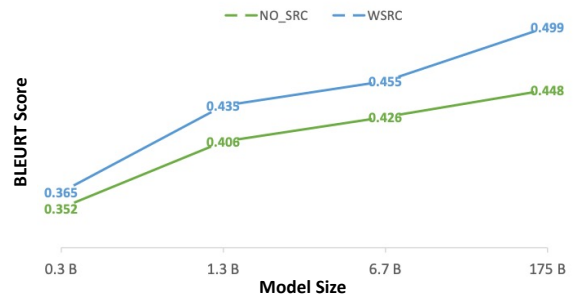


Figure 2: Average performances of various InstructGPT models based on BLEURT scores.

5.4 Human evaluation

To further validate the generated analogies more comprehensively, we also conducted human evaluation as described below.

5.4.1 Annotation Setup

We conducted the study on Amazon Mechanical Turk. Based on manual evaluation of responses to screening tests (Appendix E), we selected 17 workers for the main study.

Further, we created a sample dataset for evaluating analogies generated both in the NO_SRC and WSRC settings. In total, we generated 13k analogies⁹ in NO_SRC and 18k analogies¹⁰ in WSRC.

⁹6 analogies (5 in high temperature and 1 low temperature)

*109 target concepts*5 prompts*4 models

¹⁰6 analogies (5 in high temperature and 1 low temperature)

*109 target concepts*7 prompts*4 models

From this data, we randomly selected 42 concepts for the NO_SRC setup and 21 of them were selected for the wsrc setup (to have comparable number of analogies in both settings). The analogies for the selected concepts, generated by all the models using all the prompts in the low temperature setting were selected for evaluation since low temperature was better based on automatic evaluation.

In total, 1407 unique analogies (576 from WSRC, 770 from NO_SRC, and 61 human-generated from SAQA) were evaluated by 3 workers each, which is common in previous work on evaluation of automatically generated text (van der Lee et al., 2021). The main study had one question asking workers to evaluate whether the shown candidate analogy was meaningful for the target concept (Yes/No/Can’t decide) and provide a text input for explaining their choice (Figure 6, Appendix F). Please refer to Appendix E for more details of the study design.

5.4.2 Quantitative Results

Table 10 shows the percentage of analogies rated as meaningful, based on majority vote, for the various models and the human references from SAQA. There were <2% ties or cases with ‘Can’t decide’ as the majority, which were discarded. The Fleiss’ kappa (Fleiss, 1971) inter-annotator agreement was 0.347 in case of WSRC (plus human references for the selected concepts for wsrc concepts), indicating fair agreement and 0.553 in case of NO_SRC (plus human references for the selected concepts for NO_SRC concepts) indicating moderate agreement.

We observe that the percentage of meaningful analogies increases with model size, again confirming that larger models have a higher capacity to generate analogies. Interestingly, in the NO_SRC setting, the largest model has comparable performance to humans. We note that this doesn’t necessarily mean that those models are creative or have commonsense reasoning skills as they could have simply memorized those analogies, which a known problem of such models (Bender et al., 2021). It requires further research to test whether the models generate novel analogies unseen during training.

Moreover, upon inspection, we found that the human-generated analogies sometimes had minor issues, such as grammatical errors, which could impact their rating by annotators. So, it is possible that analogies written by experts, such as science instructors proficient in English, might be rated higher. Nevertheless, these results are quite en-

couraging as the model seems to have comparable performance to general online users who wrote the analogies in our reference dataset.

In the WSRC setting, the performance of InstructGPT is lower than human performance. This could be because there is a lesser likelihood of seeing the exact same analogy, i.e., the one asked to explain in the prompt, during training, compared to seeing *any* analogies for the target concept as required in the NO_SRC setting. So, WSRC might require more “analogical reasoning” from the models, especially for explaining analogies not seen during training. This highlights the importance of human evaluations for such tasks because otherwise, based on automatic evaluation alone, we would conclude that this is an easier setting. This is because metrics like BLEURT cannot assess the soundness of the generated reasoning.

We also compute the NO_SRC performance on the 21 shared concepts (NO_SRC₂₁, Table 10) for a fair comparison between the two settings. It is interesting to note that the performances of smaller models increase while that of larger models go down in the WSRC setting. This could be because the provided source in the prompt helps provide some guidance to the smaller models. For example, even by copying parts of the prompt (i.e., source and target), they could generate meaningful analogies (e.g., <source> is like <target>) in a few cases. Since their performance in the NO_SRC setting is very poor, even minor help or “tricks” would lead to performance improvement. On the other hand, the larger models that already performed very well, likely do not have much to gain from such help and, in fact, perform worse due to the analogical reasoning argument made above.

Overall, this highlights some limitations of the InstructGPT model for analogical reasoning, which requires further research for improvement.

5.4.3 Error Analysis

The annotators were also asked to explain their answer choice (i.e, meaningful analogy or not). By inspection, we identified the following major themes based on the workers’ explanations for choosing “not meaningful” across all models/tasks. These themes are not mutually exclusive and multiple themes were often found for one wrong generation.

1. No Analogy: This is one of the most common cases where the model failed to generate any analogy at all. Instead, it mostly generated a simple description/definition of the target concept. In a

Table 10: Percentage of meaningful analogies generated by various InstructGPT models and humans based on human evaluation. Highest value per row is underlined.

	0.3B	1.3B	6.7B	175B	Human
NO_SRC	1.90	15.61	48.29	<u>70.05</u>	66.67
WSRC	8.97	29.05	38.46	53.79	<u>71.88</u>
NO_SRC ₂₁	0	12.0	47.0	66.99	<u>71.88</u>

few cases, it also generated a tautology or an example. For example, “*The b-lymphocytes are similar to the white blood cells.*”

2. Irrelevant to target: The generated text contained little to none relevant information pertaining to the target. One interesting reason behind this was capitalization for abbreviations. For example, since the targets in the prompt were lowercased (e.g., nadh), smaller models were unable to identify abbreviations, while the larger models succeeded at this. Another reason observed was that of an ambiguous target, e.g., computer “mouse” misidentified as a rodent. In more insidious cases, the text looked correct but presented incorrect facts.

3. Incorrect source or explanation: Here, important details about the source concept were either incorrect or missing, or the provided explanation was insufficient, making the analogy completely wrong or weak at best. For example, “*A molecule of DNA is like a drop of water. It has a specific shape and size, and it can carry the genetic instructions for making a particular organism.*”

Some error types found in other natural language generations from GPT-3 (Dou et al., 2021), e.g., incoherence and grammar, were also found in our task. Further research is required to quantify them for analogical generation and attempt to fix them.

6 Limitations

A major limitation of our study is that we only studied analogies on a small reference dataset in one domain (high-school science). Our newly created reference data sets are relatively small due to limited resources found online. But, the sample size of the automatically generated analogies we evaluated was large ($\sim 31k$ automatically evaluated, and $\sim 1.4k$ manually evaluated) thereby mitigating some concerns about bias due to small dataset size. Moreover, as our research questions study an open-ended generation task, having a pre-defined list of reference candidates is not ideal for evaluation. Thus, future research is required to more thoroughly evaluate the generated analogies and investigate the generalizability of the findings to

other domains.

Further, the manual evaluation was conducted by a selected group of people in the US and might not reflect the opinions of a more diverse group. Moreover, our kappa scores of 0.3-0.5, although common in previous NLG evaluation work (van der Lee et al., 2021), are not on the higher end. In general, thresholds to determine what counts as high or low kappa scores tend to be open to interpretation (van der Lee et al., 2021). Thus, we’ve released our annotated and full datasets online, as also suggested in (van der Lee et al., 2021), and invite other researchers to further investigate them.

7 Conclusion

In this study, we proposed and studied the novel task of generating analogies by prompting InstructGPT. Our experiments showed that the InstructGPT is effective on this task when precise prompts are used, thus offering a promising new way to generate analogies, which can break the limitation of the traditional analogy generation methods in requiring a pre-generated structured representation.

By evaluating the performances of the various designed prompts in multiple temperature settings and in the presence of synthetic spelling errors, we found that the InstructGPT model is sensitive to those variations (e.g., question vs. imperative-style prompts). Additionally, based on human evaluation, we found that the quality of the generated analogies substantially depends on the model size. The largest model was found to achieve human-level performance at generating analogies for given target concepts. There is still much room for improvement at the challenging task of explaining the analogical similarity between the given target and source concepts.

Our work opens up many exciting opportunities for future work both for application-oriented and foundational research on PLMs for analogy generation. For example, conducting more robustness analyses based on prompt perturbations (e.g., natural spelling mistakes, grammar, length, etc.). Also, in addition to the unsupervised approaches we explored in this paper, it is interesting to develop supervised approaches for this task including by fine-tuning PLMs on our created datasets.

8 Acknowledgments

This work is supported in part by the IBM-Illinois Center for Cognitive Computing Systems Research (C3SR) as an IBM AI Horizon’s Network.

9 Ethical Considerations

The risks associated with using PLMs for analogy generation are similar to those of NLG tasks, such as bias, toxicity, and misinformation (Bender et al., 2021; Weidinger et al., 2021). Accordingly, these should be carefully evaluated before deploying the models for any practical applications, such as education.

Furthermore, there is a steep monetary and environmental cost associated with using the GPT-3 models, especially Davinci. The OpenAI API charges \$0.06 /1K tokens. Including early experiments, analogy generation in this study costed a total of about \$240. Since we conducted multiple runs with the same prompt account for randomness (e.g., in the high temperature setting), the costs rose sharply.

References

- Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623.
- Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.
- Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Yao Dou, Maxwell Forbes, Rik Koncel-Kedziorski, Noah A Smith, and Yejin Choi. 2021. Scarecrow: A framework for scrutinizing machine text. *arXiv preprint arXiv:2107.01294*.
- Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.
- Kenneth D Forbus, Ronald W Ferguson, Andrew Lovett, and Dedre Gentner. 2017. Extending sme to handle large-scale cognitive modeling. *Cognitive Science*, 41(5):1152–1201.
- Sai Muralidhar Jayanthi, Danish Pruthi, and Graham Neubig. 2020. Neuspell: A neural spelling correction toolkit. *arXiv preprint arXiv:2010.11085*.
- Maurice G Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.
- Aniket Kittur, Lixiu Yu, Tom Hope, Joel Chan, Hila Lifshitz-Assaf, Karni Gilon, Felicia Ng, Robert E Kraut, and Dafna Shahaf. 2019. Scaling up analogical innovation with crowds and ai. *Proceedings of the National Academy of Sciences*, 116(6):1870–1877.
- Kalpesh Krishna, Aurko Roy, and Mohit Iyyer. 2021. Hurdles to progress in long-form question answering. *arXiv preprint arXiv:2103.06332*.
- Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the second workshop on statistical machine translation*, pages 228–231.
- Junyi Li, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Pretrained language models for text generation: A survey. *arXiv preprint arXiv:2105.10311*.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.
- Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.
- Li Lucy and David Bamman. 2021. Gender and representation bias in gpt-3 generated stories. In *Proceedings of the Third Workshop on Narrative Understanding*, pages 48–55.
- Wentao Ma, Yiming Cui, Chenglei Si, Ting Liu, Shijin Wang, and Guoping Hu. 2020. Charbert: character-aware pre-trained language model. *arXiv preprint arXiv:2011.01513*.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Melanie Mitchell. 2021. Abstraction and analogy-making in artificial intelligence. *arXiv preprint arXiv:2102.10717*.
- Timothy J Newby, Peggy A Ertmer, and Donald A Stepich. 1995. Instructional analogies and the learning of concepts. *Educational Technology Research and Development*, 43(1):5–18.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al.

2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Danish Pruthi, Bhuwan Dhingra, and Zachary C Lipton. 2019. Combating adversarial misspellings with robust word recognition. *arXiv preprint arXiv:1905.11268*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Gaetano Rossiello, Alfio Gliozzo, Robert Farrell, Nicolas R Fauceglia, and Michael Glass. 2019. Learning relational representations by analogy using hierarchical siamese networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3235–3245.

Keisuke Sakaguchi, Kevin Duh, Matt Post, and Benjamin Van Durme. 2017. Robust word recognition via semi-character recurrent neural network. In *Thirty-first AAAI conference on artificial intelligence*.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of ACL*.

Peter D Turney. 2008. The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research*, 33:615–655.

Asahi Ushio, Luis Espinosa-Anke, Steven Schockaert, and Jose Camacho-Collados. 2021. Bert is to nlp what alexnet is to cv: Can pre-trained language models identify analogies? *arXiv preprint arXiv:2105.04949*.

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, and Emiel Kraemer. 2021. Human evaluation of automatically generated text: Current trends and best practice guidelines. *Computer Speech & Language*, 67:101151.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. 2021. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*.

Tony Z Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. *arXiv preprint arXiv:2102.09690*.

A Hyperparameters

Based on initial explorations, where we varied the number of maximum tokens between 0 and 1000 in

increments of 100, and then from 935-955 in increments of 1, we noticed that setting a high number of maximum tokens worked better in generating more comprehensive analogies that were not abruptly cut-off and there was little sensitivity to higher values around 950. So, we randomly chose one value in that range (939). The default value of top_p = 1 was used.

B Suitability of existing evaluation metrics

To first investigate the suitability of existing evaluation metrics for generated analogies before we can trust any evaluation results using them, we designed two testers to examine whether the existing metrics behave as expected: 1) **Ordering Tester OT**: This tester is to see if an evaluation metric can order a set of methods that have known orders between them correctly as expected. 2) **Random Perturbation Tester RPT**: This tester checks if an evaluation metric responds to a random perturbation to the ground truth data used for evaluation. A reasonable metric is expected to generate lower performance figures after perturbation.

We use those two testers to study the suitability of three popular and representative measures of automatic evaluation of generated text: BLEURT (Sellam et al., 2020), METEOR (Lavie and Agarwal, 2007), ROUGE-L (Lin, 2004).

BLEURT (B) is a recent machine learning-based metric that has been shown to capture semantic similarities between text. ROUGE-L (R)¹¹ measures longest matching subsequence of words. We use its F1-score. METEOR (M)¹² matches word stems and synonyms also.

Design of testers: We design an OT and a RPT based on the following baseline methods:

No Analogy baseline (NO_ANLGY): Here, the prompts instruct the model to generate an explanation or description of the target concept and do not ask for an analogy explicitly. Thus, we expect the generated text to be in a different “style” than analogies and the overall performance to be lower. However, the generation would still contain other relevant keywords describing the target. Thus, it is a good baseline to test if the metrics can distinguish between analogies and other descriptions.

Random baselines: For each of the three setups, we introduced random baselines

¹¹<https://pypi.org/project/rouge-score/>

¹²https://www.nltk.org/api/nltk.translate.meteor_score.html

(NO_ANLGY_RAND, NO_SRC_RAND, and WSRC_RAND, respectively) where a generated string is evaluated against a random analogy (excluding the correct matching analogy) in the reference dataset (i.e., applying a random perturbation to the ground truth). These baselines preserve the “style” of the text but not the content. We expect these methods to perform worse than their non-random counterparts.

Additionally, NO_SRC setting is expected to perform worse than WSRC because in WSRC, the model has more information (i.e., the source concept) and thus has better chances of generating the correct analogical explanation. Thus, the expected order is NO_ANLGY < NO_SRC < WSRC.

Metric testing results: Table 11 shows the overall results of experiments on the SAQA dataset using the Davinci model. Each row shows the highest average scores given by a metric in various setups (performances of each prompt are in Section 5.2 and at the end of this section.).

We can see that all the three metrics order the setups as expected, i.e., random baselines are assigned a lower score than non-random setups, and scores for NO_ANLGY < NO_SRC < WSRC. This suggests that all the three metrics have “passed” our two testers and thus can be reasonably used to evaluate whether the automatically generated analogies are similar to those generated by humans. In other words, they should help assess whether the generated text is relevant to the target concept and discuss properties of the concept that could be explained using analogies (because they passed RPT), and written in an analogical style (because they passed OT).

Moreover, the results also indicates that the InstructGPT model is able to follow the prompts in the three settings to some extent and generate non-analogical descriptions, general analogies, and analogies containing the source concepts, in those settings respectively.

In terms of discernment power, all metrics have small gaps between the scores of random and non-random settings. Similar results were previously reported in (Krishna et al., 2021) for ROUGE scores on long-form question-answering. Out of the three metrics, the BLEURT score has the largest gaps in all the settings, both between the random and non-random baselines and also between settings. It is also shown to capture semantic similarity well (Sellam et al., 2020). Thus, we use it as the main

metric in the rest of the experiments.

Table 12: Prompts for NO_ANLGY

Id	Prompt
P1	Explain <target>.
P2	What is <target>?
P3	Explain <target> in plain language to a second grader.

Table 13: Comparison of performances of different prompts and temperatures in NO_ANLGY.

	B	R	M
P1 _{tl}	0.434	0.183	0.149
P1 _{th}	0.432	0.18	0.158
P2 _{tl}	0.43	0.175	0.129
P2 _{th}	0.425	0.172	0.136
P3 _{tl}	0.445	0.180	0.132
P3 _{th}	0.444	0.179	0.144

Table 14: Comparison of performances of different prompts and temperatures in NO_SRC_RAND.

	B	R	M
P1 _{tl}	0.375	0.132	0.103
P1 _{th}	0.367	0.123	0.108
P2 _{tl}	0.359	0.116	0.092
P2 _{th}	0.366	0.127	0.105
P3 _{tl}	0.362	0.124	0.099
P3 _{th}	0.364	0.126	0.109
P4 _{tl}	0.338	0.115	0.084
P4 _{th}	0.348	0.121	0.1
P5 _{tl}	0.358	0.121	0.097
P5 _{th}	0.348	0.122	0.107

Table 15: Comparison of performances of different prompts and temperatures in WSRC_RAND.

	B	R	M
P1 _{tl}	0.37	0.120	0.094
P1 _{th}	0.363	0.122	0.107
P2 _{tl}	0.385	0.117	0.096
P2 _{th}	0.381	0.12	0.109
P3 _{tl}	0.358	0.117	0.095
P3 _{th}	0.359	0.115	0.1
P4 _{tl}	0.367	0.113	0.096
P4 _{th}	0.37	0.115	0.105
P5 _{tl}	0.36	0.113	0.09
P5 _{th}	0.356	0.117	0.094
P6 _{tl}	0.346	0.111	0.086
P6 _{th}	0.347	0.113	0.091
P7 _{tl}	0.353	0.114	0.092
P7 _{th}	0.352	0.109	0.093

Table 11: Testing results using OT and RPT. The higher score between the random baseline and the non-random setup is bolded. Highest score in a row is underlined.

	NO_ANLGY RAND	NO_ANLGY	NO_SRC RAND	NO_SRC	WSRC RAND	WSRC
B	0.349	0.445	0.375	0.462	0.385	<u>0.515</u>
R	0.122	0.183	0.132	0.196	0.122	<u>0.229</u>
M	0.099	0.158	0.109	0.171	0.109	<u>0.208</u>

Table 16: Comparison of performances of different prompts and temperatures in NO_ANLGY RAND.

	B	R	M
P1 _{tl}	0.346	0.115	0.087
P1 _{th}	0.349	0.122	0.099
P2 _{tl}	0.322	0.116	0.077
P2 _{th}	0.327	0.113	0.081
P3 _{tl}	0.334	0.111	0.079
P3 _{th}	0.336	0.11	0.081

C Experiments on STD dataset

Table 17: Prompts for STD analogies

Id	Prompt
P1	Explain <target> using an analogy.
P2	Explain <target> using a well-known analogy.
P3	What analogy is often used to explain <target>?
P4	Using a well-known analogy, explain <target>.
P5	Using an analogy, explain <target>.
P6	What is a well-known analogy to explain <target>?
P7	What is analogous to <target>?

Table 18: Most common analogies generated for each target concept in the STD dataset. #Pmt. means number of prompts that generated the shown analogy.

Target	Most common src.	# Pmt.
mind	computer	7
atom	solar system	6
heat transfer	fluid/water flow	4
sounds	wave	4
respiration	combustion	3
light	river	3
planet	rock	2
bacterial mutation	game of telephone	3
natural selection	sieve	2
gas molecules	balls	2

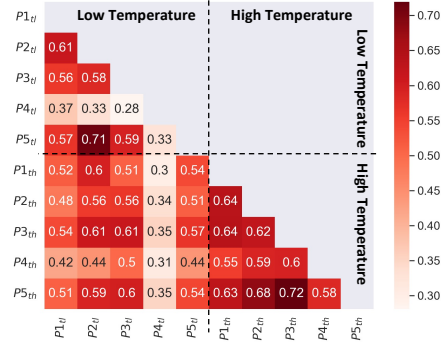


Figure 3: Kendall's Tau correlation between BLEURT scores of various prompts and temperatures in NO_SRC

D Experiments on SAQA dataset

Table 19: Comparison of lengths of generated responses by question (Q) vs. statement (S) in the WSRC setting. Question versions of the prompts generate fewer words on average, than their statement counterparts.

Prompt Pair	Avg. Len. (S)	Avg. Len. (Q)
P2-P5	43.93	34.53
P3-P6	32.55	31.4
P4-P7	42.51	32.72

Table 20: Comparison of lengths of generated responses by low and high temperatures in the NO_SRC setting. High temperature generates consistently longer analogies. Same trend is observed in other settings also.

Prompt	Avg. Length (tl)	Avg. Length (th)
P1	39.74	47.62
P2	32.67	40.71
P3	40.06	46.62
P4	32.51	40.13
P5	36.53	38.50

E Mturk study details

For identifying qualified workers on Amazon Mechanical Turk, we designed a pre-screening test (Mturk Qualification) asking them to identify the meaningful analogy for a target concept (Figure 4, Appendix F). Further, we used the following additional qualifications: workers should have com-

pleted at least 5k tasks with >98% approval rate and be located in the US since the task requires proficiency in english (this way of filtering is not perfect but there is currently no good way to identify native english speakers via Mturk). We did not collect any other demographic or geographic information about the workers.

Those who passed these qualifications worked on a small test batch of analogies asking detailed questions about their quality (Figure 5, Appendix F). The questions consisted of both Likert-style or Binary choice questions and text inputs asking them to explain their choices. We manually assessed their responses, especially paying close attention to their reasoning to identify qualified workers for the main study.

For both the main study and the screening, a simple definition of the target from sites like Simple English Wikipedia¹³ was provided to workers as reference and they were encouraged to refer to the internet to learn more about the shown concepts. We also provided several sample annotations as part of the instructions to guide workers. Moreover, we were available to answer clarification questions via a shared chatroom.

Annotators were paid at the rate of \$50/hr. The rate was decided based on open discussions with them and is above the minimum wage. They were informed that the data generated would be used for research purposes. We consulted with our university ethics board and found that IRB was not required for this study.

F Human evaluation interface

¹³https://simple.wikipedia.org/wiki/Main_Page

NADH stands for "nicotinamide adenine dinucleotide (NAD) + hydrogen (H)." This chemical occurs naturally in the body and plays a role in the chemical process that generates energy.

Which of the following options contains a good analogy to explain NADH? You may refer to the internet to learn more about NADH.

- NADH plays a role in the chemical process that generates energy.
- I always drink the energy drink NADH while playing video games with my friends.
- NADH is an important molecule in the body. NADH is kind of like a battery that stores energy until it is needed.
- NADH is like a blood vessel. It helps to carry oxygen and nutrients to your cells.
- The word "nadh" is derived from the an Egyptian word. The analogy used to explain nadh is that between a population and a tribe.

Figure 4: Pre-screening question for identifying qualified workers.

1. Do you think the text is more likely written by a human or a computer?
Please also precisely explain in a complete sentence why you chose the answer below.

Human
 Computer
 Can't decide

I chose this answer because ...

2. Does the text mention any concept as being analogous to "golgi"?

Yes
 No

3. If you answered "yes" to question 2, please write down the analogous concept.

Analogous concept

4. If you answered "yes" to question 2, does the analogy make sense to you? If you don't know about golgi or the analogous concept, please make sure to look them up on the internet to learn more about them. Please also precisely explain in a complete sentence why you chose the answer below.

Yes
 No
 Can't decide even after looking up information on the internet

I chose this answer because ...

5. If you answered "no" to question 2, please write an analogy (in ~2-3 complete sentences) for explaining golgi. Please refer to the internet if you can't think of any suitable analogies.

Analogy for golgi ...

6. To what extent does the text help explain golgi to a reader?

Very Helpful
 Somewhat Helpful
 Not so helpful
 Not helpful at all

Figure 5: Sample interface for screening qualified workers.

Background on "specialization and communication in a cell": Cell specialization is the process wherein "general" or "common" cells evolve to form specific cells that have specific functions. Cell communication is the ability of a cell to receive, process, and transmit signals with its environment and with itself.

Please carefully read the text about "specialization and communication in a cell" in the box below and answer the following questions. Pay very close attention to all the details in the text as it might contain factual errors that are hard to spot.

You are encouraged to refer to the internet if you need any additional information about "specialization and communication in a cell" or other concepts in the text.

Text:

A cell is a small, complex system that produces and receives energy. It is a part of our body and helps us to survive and thrive. A specialist in a given area, a cell can do things like make new blood or energy.

1. Does the text mention any meaningful analogy for "specialization and communication in a cell"? If you don't know about specialization and communication in a cell or the analogous concept (if any), please make sure to look them up on the internet to learn more about them. Please also precisely explain in a complete sentence why you chose the answer below.

Yes
 No
 Can't decide even after looking up information on the internet

I chose this answer because ...

Figure 6: Sample interface for human evaluation of the analogies.

Author Index

- Afyouni, Imad, 40
Alhafni, Bashar, 212
Alikhani, Malihe, 212
Alnajjar, Khalid, 100
Alonso-Moral, Jose Maria, 121
Artemova, Katya, 15
Athapaththu, Dineth, 144
Attari, Nazia, 110
- Balloccu, Simone, 156
Bhat, Suma, 29
Bhavya, Bhavya, 298
Bout, Andrey, 15
Bugarín-Diz, Alberto, 121
Burnyshev, Pavel, 15
- Chan, Chak Ho, 186
Chan, Ying-Hong, 196
Chaudhary, Amit Kumar, 288
Chen, Yue, 212
Cho, Kyunghyun, 131
Chung, Ho-Lam, 196
Cimiano, Philipp, 246
- Dobnik, Simon, 236
Déziel, Pierre-Luc, 73
- Ekanayake, Savindu Kalsara, 144
Elnagar, Ashraf, 40
Emami, Jonathan, 40
Emampoor, Yasmeen, 236
- Fan, Yao-Chung, 196
Frank, Anette, 246
- Garneau, Nicolas, 73
Gaumont, Eve, 73
González Corbelle, Javier, 121
Guerin, Frank, 225
- Han, Zhao, 1
He, He, 131
Heckmann, Martin, 110
heiko.wersing@honda-ri.de, heiko.wersing@honda-ri.de, 110
Heinisch, Philipp, 246
Hämäläinen, Mika, 100
- Ilinykh, Nikolai, 236
Inan, Mert, 212
- Jin, Yiping, 52
Jong, Maxwell, 186
- Kadam, Vishakha, 52
Karamlou, Amin, 267
Kober, Thomas H, 212
- Lamanov, Dmitry, 15
Lamontagne, Luc, 73
Lee, Joosung, 68
LI, Yucheng, 225
Lin, Chenghua, 225
Lucassen, Alex J., 288
- Mahajan, Khyati, 278
Malykh, Valentin, 15
max.koppatz@gmail.com, max.koppatz@gmail.com, 100
- Nielsen, Elizabeth Kaye, 212
Niyarepola, Kashyapa, 144
Nugues, Pierre, 40
- Opitz, Juri, 246
- Pang, Richard Yuanzhe, 131
Pfaffhauser, Marcel, 267
Piontkovskaya, Irina, 15
Poibeau, Thierry, 100
- Raji, Shahab, 212
Ranathunga, Surangika, 144
Reiter, Ehud, 156
Ros, Kevin, 186
Rygina, Polina, 1
- Sadler, Philipp, 203
Santhanam, Sashank, 278
Schlangen, David, 110, 203
Shaikh, Samira, 278
Steedman, Mark, 212
Stone, Matthew, 212
- Taboada, Juan, 121
Testoni, Alberto, 288

Tsani, Ioanna, 288

Voigt, Henrik, 203

Wanvarie, Dittaya, 52

Williams, Thomas, 1

Wootton, James R, 267

Wu, Xianze, 260

Xiong, Jinjun, 298

Yu, Yong, 260

Zarrieß, Sina, 110, 203

Zhai, ChengXiang, 186, 298

Zheng, Zaixiang, 260

Zhou, Hao, 260

Zhu, Wanzheng, 29