

BEA 2023

**18th Workshop on Innovative Use of NLP for Building  
Educational Applications**

**Proceedings of the Workshop**

July 13, 2023

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

### Gold Level



### Silver Level



©2023 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](mailto:acl@aclweb.org)

ISBN 978-1-959429-80-7

## Introduction

This year, the *Workshop on Innovative Use of NLP for Building Educational Applications* is in its 18th edition. At the same time it should be noted that, as was reminded to us by Dharmendra Kanejiya, the very first BEA workshop titled the *HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing* was run in Edmonton, Canada, in 2003, which means that this year BEA celebrates its 20th anniversary. Dharmendra presented his paper, *Automatic evaluation of students' answers using syntactically enhanced LSA*, alongside 9 other papers that were accepted to the inaugural BEA workshop. He has very fond memories of the event and highlights that he has enjoyed insightful discussions at the workshop, which back then brought together a relatively small but very important community of researchers working on educational applications using NLP, and has benefited greatly from the BEA reviewing process. Dharmendra has continued being involved in sponsoring our workshop via his company, Cognii, over a number of years, and this sponsorship has helped us support the participation of young and aspiring researchers in our workshop.

Two decades after the BEA workshop was first organized, we hope that our authors and presenters feel the same way about it as Dharmendra did and that it keeps inspiring groundbreaking work on educational applications with the use of NLP. We select our papers for acceptance on the basis of several factors, including the relevance to a core educational problem space, the novelty of the approach or domain, and the strength of the research, and, as always, excellence in research is one of the main factors considered. At the same time, the NLP field in general and our community of researchers focusing on educational applications in particular have undoubtedly grown in the past two decades: this year, we have received a record number of 110 submissions – almost twice as many as last year. From these, we have accepted 2 papers as talks, 48 as poster presentations, and 8 as system demonstrations, for an overall acceptance rate of 53 percent. Each paper was reviewed by three members of the Program Committee who we believed to be most appropriate for the paper. It is exciting to see so many excellent submissions, and we hope that with this relatively high acceptance rate we were able to include a diverse set of papers on a variety of topics and from a wide set of institutions. As in the previous years, these topics include automated writing evaluation and grading, automated item generation, reading and text complexity, educational discourse and dialogue, speech applications, grammatical error detection and correction, feedback, and educational tools and resources, among other traditional topics presented at our workshop.

At the same time, this year also marks a certain turning point in the field of NLP, with researchers starting new directions in investigating the integration and impact of Large Language Models (LLMs) on the state of the art across various tasks. The field of educational applications is no exception here: many papers that are accepted this year investigate the topics around integration of LLMs into educational applications. In addition, BEA 2023 has hosted a shared task on generation of teacher responses in educational dialogues, whose primary goal was to benchmark the ability of generative language models to act as AI teachers replying to a student in a teacher–student dialogue. Eight teams participated in this competition, and six of them have published their system description reports in our proceedings. This year, as in the previous years, we are hosting an ambassador paper talk from one of the sister societies from the International Alliance to Advance Learning in the Digital Era (IAALDE). The talk this year, titled *Generating Teacher Responses in Educational Dialogues: The AI Teacher Test*, will be given by Anaïs Tack (KU Leuven, imec). Her paper, that she will overview in this talk, received a best short paper award at EDM 2022, and the shared task is a continuation of this work.

In addition to oral, poster, and demo presentations, and the ambassador talk, BEA 2023 is hosting two keynotes. Susan Lottridge, a Chief Scientist of Natural Language Applications at Cambium Assessment, will talk about *Building Educational Applications using NLP: A Measurement Perspective*, and Jordana Heller, the Director of Data Intelligence at Textio, will talk about *Interrupting Linguistic Bias in Written Communication with NLP tools*. We are extremely grateful to our keynote speakers for agreeing to pre-

sent at our workshop and share their expertise and insights with our research community.

Last but not least, we would like to thank everyone who has been involved in organizing the BEA workshop this year. We are particularly grateful to our sponsors who keep providing their support to BEA: this year, our sponsors include Cambridge University Press & Assessment, CATALPA, Duolingo, Educational Testing Service, Grammarly, National Board of Medical Examiners, and Cognii. We would like to also thank all the authors who showed interest and submitted a paper this year. Due to the record number of submissions received, we had to extend our invitation to become part of the Program Committee to all the authors of submitted papers, and many have helped us and provided their valuable feedback and thoughtful reviews. Without this help from the community, it would not be possible to spread the reviewing load in a reasonable way, and we are very grateful to our regular reviewers as well as to emergency reviewers and all the authors who joined our PC this year and who, we hope, may become our regular PC members.

In particular, we would like to extend our gratitude to the following outstanding reviewers: Erfan Al-Hossami, Desislava Aleksandrova, Giora Alexandron, David Alfter, Alejandro Andrade, Nischal Ashok Kumar, Beata Beigman Klebanov, Marie Bexte, Abhidip Bhattacharyya, Serge Bibauw, Daniel Brenner, Chris Callison-Burch, Aubrey Condor, Steven Coyne, Sam Davidson, Jasper Degraeuwe, Thomas Demeester, Rahul Divekar and Seongjin Park, Mariano Felice, Wanyong Feng, Nigel Steven Fernandez, James Fiacco, Kotaro Funakoshi, Thomas Gaillat, Ritik Garg, Christian Gold, Nicolas Hernandez and Léane Jourdan, Joseph Marvin Imperial, Qinjin Jia, Anisia Katinskaia, Mamoru Komachi, Roland Kuhn, Alexander Kwako, Antonio Laverghetta Jr., Arun Balajiee Lekshmi Narayanan, Zhexiong Liu, Anastassia Loukina, Jiaying Lu, James H. Martin, Detmar Meurers, Phoebe Mulcaire, Ben Naismith, Sungjin Nam, Seyed Parsa Neshaei, Eda Okur, Kostiantyn Omelianchuk, Christopher Ormerod, Rebecca Passonneau, Fabio Perez, E. Margaret Perkoff, Jakob Prange, Martí Quixal, Manav Rathod, Frankie Robertson, Aiala Rosá, Igor Samokhin, Katherine Stasaski, Helmer Strik, Hakyung Sung, Abhijit Suresh, Rushil Thareja, Zhongwei Teng, Shriyash Upadhyay, Sowmya Vajjala, Justin Vasselli, Anthony Verardi, Spencer von der Ohe, Michael White, Alistair Willis, Man Fai Wong, Changrong Xiao, Kevin P. Yancey, Victoria Yaneva, Su-Youn Yoon, Roman Yangarber, Michael Zock, and Diana Galván.

Ekaterina Kochmar, MBZUAI

Jill Burstein, Duolingo

Andrea Horbach, Universität Hildesheim & CATALPA, FernUniversität in Hagen

Ronja Laarmann-Quante, Ruhr University Bochum

Nitin Madnani, Educational Testing Service

Anaïs Tack, KU Leuven, imec

Victoria Yaneva, National Board of Medical Examiners

Zheng Yuan, King's College London

Torsten Zesch, CATALPA, FernUniversität in Hagen

## **Organizers**

Ekaterina Kochmar, MBZUAI  
Jill Burstein, Duolingo  
Andrea Horbach, Universität Hildesheim & CATALPA, FernUniversität in Hagen  
Ronja Laarmann-Quante, Ruhr University Bochum  
Nitin Madnani, Educational Testing Service  
Anaïs Tack, KU Leuven, imec  
Victoria Yaneva, National Board of Medical Examiners  
Zheng Yuan, King's College London  
Torsten Zesch, CATALPA, FernUniversität in Hagen

## **Program Committee**

Sihat Anfan, Bangladesh University of Engineering and Technology  
Tazin Afrin, Educational Testing Service  
Erfan Al-Hossami, University of North Carolina at Charlotte  
Desislava Aleksandrova, CBC/Radio-Canada  
Aderajew Alem, Wachemo University  
Giora Alexandron, Weizmann Institute of Science  
David Alfter, UCLouvain  
Alejandro Andrade, Pearson  
Nischal Ashok Kumar, University of Massachusetts Amherst  
Berk Atil, Pennsylvania State University  
Rabin Banjade, University of Memphis  
Michael Gringo Angelo Bayona, Trinity College Dublin  
Lee Becker, Pearson  
Beata Beigman Klebanov, Educational Testing Service  
Marie Bexte, FernUniversität in Hagen  
Abhidip Bhattacharyya, University of Colorado Boulder  
Serge Bibauw, Universidad Central del Ecuador; UCLouvain  
Shayekh Bin Islam, Bangladesh University of Engineering and Technology  
Daniel Brenner, Educational Testing Service  
Ted Briscoe, MBZUAI  
Dominique Brunato, Institute of Computational Linguistics A. Zampolli (ILC-CNR), Pisa  
Chris Callison-Burch, University of Pennsylvania  
Jie Cao, University of Colorado  
Brian Carpenter, Indiana University of Pennsylvania  
Dumitru-Clementin Cercel, University Politehnica of Bucharest  
Chung-Chi Chen, National Institute of Advanced Industrial Science and Technology  
Guanliang Chen, Monash University  
Hyundong Cho, USC, Information Sciences Institute  
Martin Chodorow, Hunter College and the Graduate Center of CUNY  
Aubrey Condor, University of California, Berkeley  
Mark Core, University of Southern California  
Steven Coyne, Tohoku University / RIKEN  
Scott Crossley, Georgia State University  
Sam Davidson, University of California, Davis

Kordula De Kuthy, Universität Tübingen  
Jasper Degraeuwe, Ghent University  
Thomas Demeester, Ghent University - imec  
Carrie Demmans Epp, University of Alberta  
Dorottya Demszky, Stanford University  
Yuning Ding, FernUniversität in Hagen  
Rahul Divekar, Educational Testing Service  
George Duenas, Universidad Pedagógica Nacional  
Masaki Eguchi, University of Oregon/Waseda University  
Yo Ehara, Tokyo Gakugei University  
Mariano Felice, British Council  
Wanyong Feng, UMass Amherst  
Nigel Fernandez, University of Massachusetts Amherst  
James Fiacco, Carnegie Mellon University  
Michael Flor, Educational Testing Service  
Estibaliz Fraca, University College London  
Kotaro Funakoshi, Tokyo Institute of Technology  
Thomas Gaillat, Université de Rennes 2  
Ananya Ganesh, University of Colorado Boulder  
Lingyu Gao, Toyota Technological Institute at Chicago  
Rujun Gao, Texas A&M University  
Ritik Garg, Extramarks Education Pvt. Ltd.  
Christian Gold, FernUniversität in Hagen  
Samuel González-López, Technological University of Nogales  
Le An Ha, RGCL, RIILP, University of Wolverhampton  
Ching Nam Hang, Department of Computer Science, City University of Hong Kong  
Nicolas Hernandez, Nantes University  
Chung-Chi Huang, Frostburg State University  
Ping-Yu Huang, Ming Chi University of Technology  
Yi-Ting Huang, Academia Sinica  
David Huggins-Daines, Independent Researcher  
Yusuke Ide, Nara Institute of Science and Technology  
Joseph Marvin Imperial, University of Bath; National University Philippines  
Radu Tudor Ionescu, University of Bucharest  
Qinjin Jia, North Carolina State University  
Helen Jin, University of Pennsylvania  
Richard Johansson, University of Gothenburg  
Masahiro Kaneko, Tokyo Institute of Technology  
Neha Kardam, University of Washington  
Anisia Katinskaia, University of Helsinki  
Elma Kerz, RWTH Aachen University  
Mamoru Komachi, Hitotsubashi University  
Roland Kuhn, National Research Council of Canada  
Alexander Kwako, University of California, Los Angeles  
Kristopher Kyle, University of Oregon  
Geoffrey LaFlair, Duolingo  
Antonio Laverghetta Jr., University of South Florida  
Jaewook Lee, UMass Amherst  
Ji-Ung Lee, UKP, TU Darmstadt  
Arun Balajiee Lekshmi Narayanan, University of Pittsburgh  
Xu Li, Zhejiang University

Chengyuan Liu, North Carolina State University  
Yudong Liu, Western Washington University  
Zhexiong Liu, University of Pittsburgh  
Zoey Liu, Department of Linguistics, University of Florida  
Susan Lottridge, Cambium Assessment  
Anastassia Loukina, Grammarly Inc  
Jiaying Lu, Emory University  
Jakub Macina, ETH Zurich  
Lieve Macken, Ghent University  
James H. Martin, University of Colorado Boulder  
Sandeep Mathias, Presidency University  
Janet Mee, National Board of Medical Examiners  
Detmar Meurers, Universität Tübingen  
Phoebe Mulcaire, Duolingo  
Tsegay Mullu, Wachemo University  
Faizan E Mustafa, QUIBIQ GmbH  
Farah Nadeem, World Bank  
Ben Naismith, Duolingo  
Sungjin Nam, ACT, Inc  
Diane Napolitano, The Associated Press  
Kamel Nebhi, Education First  
Seyed Parsa Neshaei, Sharif University of Technology  
Hwee Tou Ng, National University of Singapore  
Huy Nguyen, Amazon  
Gebregziabihier Nigusie, Mizan-Tepi University  
S Jaya Nirmala, National Institute of Technology Tiruchirappalli  
Kai North, George Mason University  
Eda Okur, Intel Labs  
Priti Oli, University of Memphis  
Kostiantyn Omelianchuk, Grammarly  
Brian Ondov, National Library of Medicine  
Christopher Ormerod, Cambium Assessment  
Simon Ostermann, German Research Center for Artificial Intelligence (DFKI)  
Ulrike Pado, HFT Stuttgart  
Frank Palma Gomez, City University of New York, Queens College  
Chanjun Park, Upstage  
Rebecca Passonneau, The Pennsylvania State University  
Fabio Perez, Independent Researcher  
E. Margaret Perkoff, University of Colorado Boulder  
Jakob Prange, Hong Kong Polytechnic University  
Reinald Adrian Pugoy, University of the Philippines Open University  
Long Qin, Alibaba  
Mengyang Qiu, University at Buffalo  
Martí Quixal, University of Tübingen  
Arjun Ramesh Rao, Microsoft  
Vivi Rantung, Universitas Negeri Manado  
Manav Rathod, Glean  
Brian Riordan, Educational Testing Service  
Frankie Robertson, University of Jyväskylä  
Aiala Rosá, Instituto de Computación, Facultad de Ingeniería, Universidad de la República  
Carolyn Rosé, Carnegie Mellon University



Alla Rozovskaya, Queens College, City University of New York  
Igor Samokhin, Grammarly  
Alexander Scarlatos, University of Massachusetts Amherst  
Matthew Shardlow, Manchester Metropolitan University  
Anchal Sharma, PES University  
Shady Shehata, Mohamed bin Zayed University of Artificial Intelligence (MBZUAI)  
Gyu-Ho Shin, Palacký University Olomouc  
Shashank Sonkar, Rice University  
Katherine Stasaski, Salesforce Research  
Helmer Strik, Centre for Language and Speech Technology (CLST), Centre for Language Studies (CLS), Radboud University Nijmegen  
Hakyung Sung, University of Oregon  
Abhijit Suresh, Reddit Inc.  
Xiangru Tang, Yale University  
Zhongwei Teng, Vanderbilt University  
Rushil Thareja, Extramarks Education Pvt. Ltd.  
Naveen Thomas, Texas A&M University  
Alexandra Uitdenbogerd, RMIT University  
Shriyash Upadhyay, Martian  
Masaki Uto, The University of Electro-Communications  
Sowmya Vajjala, National Research Council  
Justin Vasselli, Nara Institute of Science and Technology  
Giulia Venturi, Institute of Computational Linguistics Antonio Zampolli (ILC-CNR)  
Anthony Verardi, Duolingo  
Carl Vogel, Trinity College Dublin  
Elena Volodina, University of Gothenburg  
Spencer Von Der Ohe, University of Alberta  
Zichao Wang, Adobe Research  
Taro Watanabe, Nara Institute of Science and Technology  
Michael White, The Ohio State University  
Alistair Willis, The Open University  
Man Fai Wong, City University of Hong Kong  
Menbere Worku, Wachemo University  
Changrong Xiao, Tsinghua University  
Yiqiao Xu, North Carolina State University  
Kevin P. Yancey, Duolingo  
Roman Yangarber, University of Helsinki  
Su-Youn Yoon, EduLab  
Kamyar Zeinalipour, University of Siena  
Jing Zhang, Emory University  
Hengyuan Zhang, Tsinghua University  
Jessica Zipf, University of Konstanz  
Michael Zock, CNRS-LIS  
Jan Švec, NTIS, University of West Bohemia

# Keynote Talk: Building Educational Applications using NLP: A Measurement Perspective

**Susan Lottridge**  
Cambium Assessment

**Abstract:** The domains of NLP, data science, software engineering, and educational measurement are becoming increasingly interdependent when creating NLP-based educational applications. Indeed, the domains themselves are merging in key ways, with each incorporating one another's methods and tools into their work. For example, many software engineers regularly deploy machine learning models and many linguists, data scientists, and measurement staff regularly develop software. Even so, each discipline approaches this complex task with the assumptions, priorities, and values of their field. The best educational applications are the result of multi-disciplinary teams that can leverage one another's strengths and can recognize and honor the values of each disciplinary perspective.

This talk will describe the educational measurement perspective within this collaborative process. At a high level, educational measurement is the design, use, and analysis of assessments in order to make inferences about what students know and can do. Given this, the measurement experts on a team focus heavily on defining what students need to know and do, what evidence supports inferences about what students know and can do, and whether the data are accurate, reliable, and fair to all students. This perspective can impact the full life-cycle development of educational applications, from designing the core product focus, data collection activities, NLP modelling, analysis of model outputs, and information provided to students. It can also help ensure that educational applications produce information that is valuable to teachers and students. Because these perspectives can be opaque to those outside of measurement, the development process of various NLP educational tools will be used to illustrate key areas where measurement can contribute in product design.

**Bio:** Sue Lottridge is a Chief Scientist of Natural Language Applications at Cambium Assessment, Inc. She has a Ph.D. in Assessment and Measurement from James Madison University and Masters' degrees in Mathematics and Computer Science from the University of Wisconsin – Madison. In this role, she leads CAI's machine learning and scoring team on the research, development, and operation of CAI's automated scoring and feedback software. Dr. Lottridge has worked in automated scoring for fifteen years and has contributed to the design, research, and use of multiple automated scoring engines including equation scoring, essay scoring, short answer scoring, speech scoring, crisis alert detection, and essay feedback.

# Keynote Talk: Interrupting Linguistic Bias in Written Communication with NLP tools

Jordana Heller

Textio

**Abstract:** Unconscious bias is hard to detect, but when we identify it in language usage, we can take steps to interrupt and reduce it. At Textio, we focus on using NLP to detect, interrupt, and educate writers about bias in written workforce communications. Unconscious bias affects many facets of the employee lifecycle. Exclusionary language in recruiting communications can deter candidates from diverse backgrounds from even applying to a position, hindering efforts to build inclusive workplaces. Once a candidate has accepted a position, the language used to provide them feedback on their performance affects how they develop professionally, and we have found stark inequities in the language of feedback to members of different demographic groups. This talk will discuss how Textio uses NLP to interrupt these patterns of bias by assessing these texts for bias and providing 1) real-time iterative, educational feedback to the writer on how to improve a specific document, including guidance toward less-biased language alternatives, and 2) an assessment at a workplace level of exclusionary and inclusive language, so that companies can set goals around language improvement and track their progress toward them.

**Bio:** Jordana Heller, PhD, is Director of Data Intelligence at Textio, a tech company focused on interrupting bias in performance feedback and recruiting. Textio identifies bias in written documents and provides data to writers in real time that helps them write more effectively and equitably. At Textio, Jordana applies her background as a computational psycholinguist and cognitive scientist to her leadership of R&D teams who are focused on using data and NLP to help employers reduce bias and accelerate professional growth equitably.

# Keynote Talk: Generating Teacher Responses in Educational Dialogues: The AI Teacher Test & BEA 2023 Shared Task

Anaïs Tack

KU Leuven, imec

Ambassador paper presentation from the 15th International Conference on Educational Data Mining (EDM 2022), a member society of the IAALDE (International Alliance to Advance Learning in the Digital Era)

**Abstract:** How can we test whether state-of-the-art generative models, such as Blender and GPT-3, are good AI teachers, capable of replying to a student in an educational dialogue? Designing an AI teacher test is challenging: although evaluation methods are much-needed, there is no off-the-shelf solution to measuring pedagogical ability.

In the first part of this talk, I will describe our paper *The AI Teacher Test: Measuring the Pedagogical Ability of Blender and GPT-3 in Educational Dialogues* presented at EDM 2022. The paper reported on a first attempt at an AI teacher test. We built a solution around the insight that you can run conversational agents in parallel to human teachers in real-world dialogues, simulate how different agents would respond to a student, and compare these counterpart responses in terms of three abilities: speak like a teacher, understand a student, help a student. Our method builds on the reliability of comparative judgments in education and uses a probabilistic model and Bayesian sampling to infer estimates of pedagogical ability. We find that, even though conversational agents (Blender in particular) perform well on conversational uptake, they are quantifiably worse than real teachers on several pedagogical dimensions, especially with regard to helpfulness.

In the second part of this talk, I will describe the results of the *BEA 2023 Shared Task on Generating AI Teacher Responses in Educational Dialogues*, which was a continuation of our EDM paper.

**Bio:** Anaïs Tack is a postdoctoral researcher working on language technology for smart education at itec, an imec research group at KU Leuven, and is also a lecturer in NLP at UCLouvain. She holds a joint Ph.D. in linguistics from UCLouvain and KU Leuven, where she worked as an F.R.S.-FNRS doctoral research fellow. She was a BAEF postdoctoral scholar and research fellow at Stanford University, where she worked in Chris Piech's lab and the Stanford HAI education team. Her research interests include the generation and evaluation of teacher language in educational dialogues, the prediction of lexical difficulty for non-native readers, the automated scoring of language proficiency for non-native writers, and the creation of machine-readable resources from educational materials. Anaïs participated in organizing the CWI shared task at BEA 2018 as well as the 27th International EUROCALL conference in 2019. She is an executive board member of the ACL SIGEDU and has been involved in organizing the BEA workshop since 2021.

## Table of Contents

<i>LFTK: Handcrafted Features in Computational Linguistics</i> Bruce W. Lee and Jason Lee .....	1
<i>Improving Mathematics Tutoring With A Code Scratchpad</i> Shriyash Upadhyay, Etan Ginsberg and Chris Callison-Burch .....	20
<i>A Transfer Learning Pipeline for Educational Resource Discovery with Application in Survey Generation</i> Irene Li, Thomas George, Alex Fabbri, Tammy Liao, Benjamin Chen, Rina Kawamura, Richard Zhou, Vanessa Yan, Swapnil Hingmire and Dragomir Radev .....	29
<i>Using Learning Analytics for Adaptive Exercise Generation</i> Tanja Heck and Detmar Meurers .....	44
<i>Reviewwriter: AI-Generated Instructions For Peer Review Writing</i> Xiaotian Su, Thiemo Wambsganss, Roman Rietsche, Seyed Parsa Neshaei and Tanja Kser . . . .	57
<i>Towards L2-friendly pipelines for learner corpora: A case of written production by L2-Korean learners</i> Hakyung Sung and Gyu-Ho Shin .....	72
<i>ChatBack: Investigating Methods of Providing Grammatical Error Feedback in a GUI-based Language Learning Chatbot</i> Kai-Hui Liang, Sam Davidson, Xun Yuan, Shehan Panditharatne, Chun-Yen Chen, Ryan Shea, Derek Pham, Yinghua Tan, Erik Voss and Luke Fryer .....	83
<i>Enhancing Video-based Learning Using Knowledge Tracing: Personalizing Students' Learning Experience with ORBITS</i> Shady Shehata, David Santandreu Calonge, Philip Purnell and Mark Thompson .....	100
<i>Enhancing Human Summaries for Question-Answer Generation in Education</i> Hannah Gonzalez, Liam Dugan, Eleni Miltsakaki, Zhiqi Cui, Jiaxuan Ren, Bryan Li, Shriyash Upadhyay, Etan Ginsberg and Chris Callison-Burch .....	108
<i>Difficulty-Controllable Neural Question Generation for Reading Comprehension using Item Response Theory</i> Masaki Uto, Yuto Tomikawa and Ayaka Suzuki .....	119
<i>Evaluating Classroom Potential for Card-it: Digital Flashcards for Studying and Learning Italian Morphology</i> Mariana Shimabukuro, Jessica Zipf, Shawn Yama and Christopher Collins .....	130
<i>Scalable and Explainable Automated Scoring for Open-Ended Constructed Response Math Word Problems</i> Scott Hellman, Alejandro Andrade and Kyle Habermehl .....	137
<i>Gender-Inclusive Grammatical Error Correction through Augmentation</i> Gunnar Lund, Kostiantyn Omelianchuk and Igor Samokhin .....	148
<i>ReadAlong Studio Web Interface for Digital Interactive Storytelling</i> Aidan Pine, David Huggins-Daines, Eric Joanis, Patrick Littell, Marc Tessier, Delasie Torkornoo, Rebecca Knowles, Roland Kuhn and Delaney Lothian .....	163

<i>Labels are not necessary: Assessing peer-review helpfulness using domain adaptation based on self-training</i>	
Chengyuan Liu, Divyang Doshi, Muskaan Bhargava, Ruixuan Shang, Jialin Cui, Dongkuan Xu and Edward Gehringer . . . . .	173
<i>Generating Dialog Responses with Specified Grammatical Items for Second Language Learning</i>	
Yuki Okano, Kotaro Funakoshi, Ryo Nagata and Manabu Okumura . . . . .	184
<i>UKP-SQuARE: An Interactive Tool for Teaching Question Answering</i>	
Haishuo Fang, Haritz Puerto and Iryna Gurevych . . . . .	195
<i>Exploring Effectiveness of GPT-3 in Grammatical Error Correction: A Study on Performance and Controllability in Prompt-Based Methods</i>	
Mengsay Loem, Masahiro Kaneko, Sho Takase and Naoaki Okazaki . . . . .	205
<i>A Closer Look at k-Nearest Neighbors Grammatical Error Correction</i>	
Justin Vasselli and Taro Watanabe . . . . .	220
<i>Towards Extracting and Understanding the Implicit Rubrics of Transformer Based Automatic Essay Scoring Models</i>	
James Fiacco, David Adamson and Carolyn Ros . . . . .	232
<i>Analyzing Bias in Large Language Model Solutions for Assisted Writing Feedback Tools: Lessons from the Feedback Prize Competition Series</i>	
Perpetual Baffour, Tor Saxberg and Scott Crossley . . . . .	242
<i>Improving Reading Comprehension Question Generation with Data Augmentation and Overgenerate-and-rank</i>	
Nischal Ashok Kumar, Nigel Fernandez, Zichao Wang and Andrew Lan . . . . .	247
<i>Assisting Language Learners: Automated Trans-Lingual Definition Generation via Contrastive Prompt Learning</i>	
Hengyuan Zhang, Dawei Li, Yanran Li, Chenming Shang, Chufan Shi and Yong Jiang . . . . .	260
<i>Predicting the Quality of Revisions in Argumentative Writing</i>	
Zhexiong Liu, Diane Litman, Elaine Wang, Lindsay Matsumura and Richard Correnti . . . . .	275
<i>Reconciling Adaptivity and Task Orientation in the Student Dashboard of an Intelligent Language Tutoring System</i>	
Leona Colling, Tanja Heck and Detmar Meurers . . . . .	288
<i>GrounDialog: A Dataset for Repair and Grounding in Task-oriented Spoken Dialogues for Language Learning</i>	
Xuanming Zhang, Rahul Divekar, Rutuja Ubale and Zhou Yu . . . . .	300
<i>SIGHT: A Large Annotated Dataset on Student Insights Gathered from Higher Education Transcripts</i>	
Rose Wang, Pawan Wirawarn, Noah Goodman and Dorottya Demszky . . . . .	315
<i>Recognizing Learner Handwriting Retaining Orthographic Errors for Enabling Fine-Grained Error Feedback</i>	
Christian Gold, Ronja Laarmann-Quante and Torsten Zesch . . . . .	352
<i>ExASAG: Explainable Framework for Automatic Short Answer Grading</i>	
Maximilian Tornqvist, Mosleh Mahamud, Erick Mendez Guzman and Alexandra Farazouli . . . . .	361
<i>You've Got a Friend in ... a Language Model? A Comparison of Explanations of Multiple-Choice Items of Reading Comprehension between ChatGPT and Humans</i>	
George Duenas, Sergio Jimenez and Geral Mateus Ferro . . . . .	372

<i>Automatically Generated Summaries of Video Lectures May Enhance Students' Learning Experience</i> Hannah Gonzalez, Jiening Li, Helen Jin, Jiaxuan Ren, Hongyu Zhang, Ayotomiwa Akinyele, Adrian Wang, Eleni Miltsakaki, Ryan Baker and Chris Callison-Burch . . . . .	382
<i>Automated evaluation of written discourse coherence using GPT-4</i> Ben Naismith, Phoebe Mulcaire and Jill Burstein . . . . .	394
<i>ALEXSIS+: Improving Substitute Generation and Selection for Lexical Simplification with Information Retrieval</i> Kai North, Alphaeus Dmonte, Tharindu Ranasinghe, Matthew Shardlow and Marcos Zampieri 404	
<i>Generating Better Items for Cognitive Assessments Using Large Language Models</i> Antonio Laverghetta Jr. and John Licato . . . . .	414
<i>Span Identification of Epistemic Stance-Taking in Academic Written English</i> Masaki Eguchi and Kristopher Kyle . . . . .	429
<i>ACTA: Short-Answer Grading in High-Stakes Medical Exams</i> King Yiu Suen, Victoria Yaneva, Le An Ha, Janet Mee, Yiyun Zhou and Polina Harik . . . . .	443
<i>Hybrid Models for Sentence Readability Assessment</i> Fengkai Liu and John Lee . . . . .	448
<i>Training for Grammatical Error Correction Without Human-Annotated L2 Learners' Corpora</i> Mikio Oda . . . . .	455
<i>Exploring a New Grammatico-functional Type of Measure as Part of a Language Learning Expert System</i> Cyriel Mallart, Andrew Simpkin, Rmi Venant, Nicolas Ballier, Bernardo Stearns, Jen Yu Li and Thomas Gaillat . . . . .	466
<i>Japanese Lexical Complexity for Non-Native Readers: A New Dataset</i> Yusuke Ide, Masato Mita, Adam Nohejl, Hiroki Ouchi and Taro Watanabe . . . . .	477
<i>Grammatical Error Correction for Sentence-level Assessment in Language Learning</i> Anisia Katinskaia and Roman Yangarber . . . . .	488
<i>Geen makkie: Interpretable Classification and Simplification of Dutch Text Complexity</i> Eliza Hobo, Charlotte Pouw and Lisa Beinborn . . . . .	503
<i>CEFR-based Contextual Lexical Complexity Classifier in English and French</i> Desislava Aleksandrova and Vincent Pouliot . . . . .	518
<i>The NCTE Transcripts: A Dataset of Elementary Math Classroom Transcripts</i> Dorottya Demszky and Heather Hill . . . . .	528
<i>Auto-req: Automatic detection of pre-requisite dependencies between academic videos</i> Rushil Thareja, Ritik Garg, Shiva Baghel, Deep Dwivedi, Mukesh Mohania and Ritvik Kulshre- stha . . . . .	539
<i>Transformer-based Hebrew NLP models for Short Answer Scoring in Biology</i> Abigail Gurin Schleifer, Beata Beigman Klebanov, Moriah Ariely and Giora Alexandron . . . . .	550
<i>Comparing Neural Question Generation Architectures for Reading Comprehension</i> E. Margaret Perkoff, Abhidip Bhattacharyya, Jon Cai and Jie Cao . . . . .	556

<i>A dynamic model of lexical experience for tracking of oral reading fluency</i> Beata Beigman Klebanov, Michael Suhan, Zuowei Wang and Tenaha O’reilly .....	567
<i>Rating Short L2 Essays on the CEFR Scale with GPT-4</i> Kevin P. Yancey, Geoffrey Laflair, Anthony Verardi and Jill Burstein .....	576
<i>Towards automatically extracting morphosyntactical error patterns from L1-L2 parallel dependency treebanks</i> Arianna Masciolini, Elena Volodina and Dana Dannlls .....	585
<i>Learning from Partially Annotated Data: Example-aware Creation of Gap-filling Exercises for Language Learning</i> Semere Kiros Bitew, Johannes Deleu, A. Seza Doruz, Chris Develder and Thomas Demeester	598
<i>Evaluating Reading Comprehension Exercises Generated by LLMs: A Showcase of ChatGPT in Education Applications</i> Changrong Xiao, Sean Xin Xu, Kunpeng Zhang, Yufang Wang and Lei Xia .....	610
<i>Is ChatGPT a Good Teacher Coach? Measuring Zero-Shot Performance For Scoring and Providing Actionable Insights on Classroom Instruction</i> Rose Wang and Dorottya Demszky .....	626
<i>Does BERT Exacerbate Gender or L1 Biases in Automated English Speaking Assessment?</i> Alexander Kwako, Yixin Wan, Jieyu Zhao, Mark Hansen, Kai-Wei Chang and Li Cai .....	668
<i>MultiQG-TI: Towards Question Generation from Multi-modal Sources</i> Zichao Wang and Richard Baraniuk .....	682
<i>Inspecting Spoken Language Understanding from Kids for Basic Math Learning at Home</i> Eda Okur, Roddy Fuentes Alba, Saurav Sahay and Lama Nachman .....	692
<i>Socratic Questioning of Novice Debuggers: A Benchmark Dataset and Preliminary Evaluations</i> Erfan Al-Hossami, Razvan Bunescu, Ryan Teehan, Laurel Powell, Khyati Mahajan and Mohsen Dorodchi .....	709
<i>Beyond Black Box AI generated Plagiarism Detection: From Sentence to Document Level</i> Ali Quidwai, Chunhui Li and Parijat Dube .....	727
<i>Enhancing Educational Dialogues: A Reinforcement Learning Approach for Generating AI Teacher Responses</i> Thomas Huber, Christina Niklaus and Siegfried Handschuh .....	736
<i>Assessing the efficacy of large language models in generating accurate teacher responses</i> Yann Hicke, Abhishek Masand, Wentao Guo and Tushaar Gangavarapu .....	745
<i>RETUYT-InCo at BEA 2023 Shared Task: Tuning Open-Source LLMs for Generating Teacher Responses</i> Alexis Baladn, Ignacio Sastre, Luis Chiruzzo and Aiala Ros .....	756
<i>Empowering Conversational Agents using Semantic In-Context Learning</i> Amin Omidvar and Aijun An .....	766
<i>NAISTeacher: A Prompt and Rerank Approach to Generating Teacher Utterances in Educational Dialogues</i> Justin Vasselli, Christopher Vasselli, Adam Nohejl and Taro Watanabe .....	772
<i>The BEA 2023 Shared Task on Generating AI Teacher Responses in Educational Dialogues</i> Anas Tack, Ekaterina Kochmar, Zheng Yuan, Serge Bibauw and Chris Piech .....	785



*The ADAIO System at the BEA-2023 Shared Task: Shared Task Generating AI Teacher Responses in Educational Dialogues*  
Adaeze Adigwe and Zheng Yuan ..... 796

# Program

**Thursday, July 13, 2023**

- 09:00 - 09:05     *Opening Remarks*
- 09:05 - 09:50     *Keynote by Susan Lottridge (Cambium Assessment). 'Building Educational Applications using NLP: A Measurement Perspective'*
- 09:50 - 10:30     *Outstanding Papers*
- Improving Reading Comprehension Question Generation with Data Augmentation and Overgenerate-and-rank*  
Nischal Ashok Kumar, Nigel Fernandez, Zichao Wang and Andrew Lan
- Grammatical Error Correction for Sentence-level Assessment in Language Learning*  
Anisia Katinskaia and Roman Yangarber
- 10:30 - 11:00     *Morning Coffee Break*
- 11:00 - 11:30     *Spotlight talks for Poster / Demo Session A (In-person + Virtual)*
- 11:30 - 12:30     *Poster / Demo Session*
- LFTK: Handcrafted Features in Computational Linguistics*  
Bruce W. Lee and Jason Lee
- A Transfer Learning Pipeline for Educational Resource Discovery with Application in Survey Generation*  
Irene Li, Thomas George, Alex Fabbri, Tammy Liao, Benjamin Chen, Rina Kawamura, Richard Zhou, Vanessa Yan, Swapnil Hingmire and Dragomir Radev
- Using Learning Analytics for Adaptive Exercise Generation*  
Tanja Heck and Detmar Meurers
- Reviewriter: AI-Generated Instructions For Peer Review Writing*  
Xiaotian Su, Thiemo Wambsganss, Roman Rietsche, Seyed Parsa Neshaei and Tanja Kser
- Towards L2-friendly pipelines for learner corpora: A case of written production by L2-Korean learners*  
Hakyung Sung and Gyu-Ho Shin

**Thursday, July 13, 2023 (continued)**

*ChatBack: Investigating Methods of Providing Grammatical Error Feedback in a GUI-based Language Learning Chatbot*

Kai-Hui Liang, Sam Davidson, Xun Yuan, Shehan Panditharatne, Chun-Yen Chen, Ryan Shea, Derek Pham, Yinghua Tan, Erik Voss and Luke Fryer

*Enhancing Video-based Learning Using Knowledge Tracing: Personalizing Students' Learning Experience with ORBITS*

Shady Shehata, David Santandreu Calonge, Philip Purnell and Mark Thompson

*Enhancing Human Summaries for Question-Answer Generation in Education*

Hannah Gonzalez, Liam Dugan, Eleni Miltsakaki, Zhiqi Cui, Jiaxuan Ren, Bryan Li, Shriyash Upadhyay, Etan Ginsberg and Chris Callison-Burch

*Difficulty-Controllable Neural Question Generation for Reading Comprehension using Item Response Theory*

Masaki Uto, Yuto Tomikawa and Ayaka Suzuki

*Evaluating Classroom Potential for Card-it: Digital Flashcards for Studying and Learning Italian Morphology*

Mariana Shimabukuro, Jessica Zipf, Shawn Yama and Christopher Collins

*Gender-Inclusive Grammatical Error Correction through Augmentation*

Gunnar Lund, Kostiantyn Omelianchuk and Igor Samokhin

*Labels are not necessary: Assessing peer-review helpfulness using domain adaptation based on self-training*

Chengyuan Liu, Divyang Doshi, Muskaan Bhargava, Ruixuan Shang, Jialin Cui, Dongkuan Xu and Edward Gehring

*Generating Dialog Responses with Specified Grammatical Items for Second Language Learning*

Yuki Okano, Kotaro Funakoshi, Ryo Nagata and Manabu Okumura

*Exploring Effectiveness of GPT-3 in Grammatical Error Correction: A Study on Performance and Controllability in Prompt-Based Methods*

Mengsay Loem, Masahiro Kaneko, Sho Takase and Naoaki Okazaki

*A Closer Look at k-Nearest Neighbors Grammatical Error Correction*

Justin Vasselli and Taro Watanabe

*Analyzing Bias in Large Language Model Solutions for Assisted Writing Feedback Tools: Lessons from the Feedback Prize Competition Series*

Perpetual Baffour, Tor Saxberg and Scott Crossley

Thursday, July 13, 2023 (continued)

*Assisting Language Learners: Automated Trans-Lingual Definition Generation via Contrastive Prompt Learning*

Hengyuan Zhang, Dawei Li, Yanran Li, Chenming Shang, Chufan Shi and Yong Jiang

*Predicting the Quality of Revisions in Argumentative Writing*

Zhexiong Liu, Diane Litman, Elaine Wang, Lindsay Matsumura and Richard Correnti

*Reconciling Adaptivity and Task Orientation in the Student Dashboard of an Intelligent Language Tutoring System*

Leona Colling, Tanja Heck and Detmar Meurers

*GrounDialog: A Dataset for Repair and Grounding in Task-oriented Spoken Dialogues for Language Learning*

Xuanming Zhang, Rahul Divekar, Rutuja Ubale and Zhou Yu

*SIGHT: A Large Annotated Dataset on Student Insights Gathered from Higher Education Transcripts*

Rose Wang, Pawan Wirawarn, Noah Goodman and Dorottya Demszky

*Recognizing Learner Handwriting Retaining Orthographic Errors for Enabling Fine-Grained Error Feedback*

Christian Gold, Ronja Laarmann-Quante and Torsten Zesch

*ExASAG: Explainable Framework for Automatic Short Answer Grading*

Maximilian Tornqvist, Mosleh Mahamud, Erick Mendez Guzman and Alexandra Farazouli

*You've Got a Friend in ... a Language Model? A Comparison of Explanations of Multiple-Choice Items of Reading Comprehension between ChatGPT and Humans*

George Duenas, Sergio Jimenez and Geral Mateus Ferro

*Automatically Generated Summaries of Video Lectures May Enhance Students' Learning Experience*

Hannah Gonzalez, Jiening Li, Helen Jin, Jiaxuan Ren, Hongyu Zhang, Ayotomiwa Akinyele, Adrian Wang, Eleni Miltsakaki, Ryan Baker and Chris Callison-Burch

*Span Identification of Epistemic Stance-Taking in Academic Written English*

Masaki Eguchi and Kristopher Kyle

*Hybrid Models for Sentence Readability Assessment*

Fengkai Liu and John Lee

**Thursday, July 13, 2023 (continued)**

*Geen makkie: Interpretable Classification and Simplification of Dutch Text Complexity*

Eliza Hobo, Charlotte Pouw and Lisa Beinborn

*Towards automatically extracting morphosyntactical error patterns from L1-L2 parallel dependency treebanks*

Arianna Masciolini, Elena Volodina and Dana Dannlls

*Learning from Partially Annotated Data: Example-aware Creation of Gap-filling Exercises for Language Learning*

Semere Kiros Bitew, Johannes Deleu, A. Seza Doruz, Chris Develder and Thomas Demeester

*Beyond Black Box AI generated Plagiarism Detection: From Sentence to Document Level*

Ali Quidwai, Chunhui Li and Parijat Dube

*Enhancing Educational Dialogues: A Reinforcement Learning Approach for Generating AI Teacher Responses*

Thomas Huber, Christina Niklaus and Siegfried Handschuh

12:30 - 14:00 *Lunch Break*

14:00 - 14:30 *Spotlight talks for Poster / Demo Session B (In-person + Virtual)*

14:30 - 15:30 *Posters / Demo Session*

*Improving Mathematics Tutoring With A Code Scratchpad*

Shriyash Upadhyay, Etan Ginsberg and Chris Callison-Burch

*Scalable and Explainable Automated Scoring for Open-Ended Constructed Response Math Word Problems*

Scott Hellman, Alejandro Andrade and Kyle Habermehl

*ReadAlong Studio Web Interface for Digital Interactive Storytelling*

Aidan Pine, David Huggins-Daines, Eric Joanis, Patrick Littell, Marc Tessier, Delasie Torkornoo, Rebecca Knowles, Roland Kuhn and Delaney Lothian

*Labels are not necessary: Assessing peer-review helpfulness using domain adaptation based on self-training*

Chengyuan Liu, Divyang Doshi, Muskaan Bhargava, Ruixuan Shang, Jialin Cui, Dongkuan Xu and Edward Gehring

**Thursday, July 13, 2023 (continued)**

*UKP-SQuARE: An Interactive Tool for Teaching Question Answering*

Haishuo Fang, Haritz Puerto and Iryna Gurevych

*Towards Extracting and Understanding the Implicit Rubrics of Transformer Based Automatic Essay Scoring Models*

James Fiacco, David Adamson and Carolyn Ros

*You've Got a Friend in ... a Language Model? A Comparison of Explanations of Multiple-Choice Items of Reading Comprehension between ChatGPT and Humans*

George Duenas, Sergio Jimenez and Geral Mateus Ferro

*Automated evaluation of written discourse coherence using GPT-4*

Ben Naismith, Phoebe Mulcaire and Jill Burstein

*ALEXSIS+: Improving Substitute Generation and Selection for Lexical Simplification with Information Retrieval*

Kai North, Alphaeus Dmonte, Tharindu Ranasinghe, Matthew Shardlow and Marcos Zampieri

*Generating Better Items for Cognitive Assessments Using Large Language Models*

Antonio Laverghetta Jr. and John Licato

*ACTA: Short-Answer Grading in High-Stakes Medical Exams*

King Yiu Suen, Victoria Yaneva, Le An Ha, Janet Mee, Yiyun Zhou and Polina Harik

*Training for Grammatical Error Correction Without Human-Annotated L2 Learners' Corpora*

Mikio Oda

*Exploring a New Grammatico-functional Type of Measure as Part of a Language Learning Expert System*

Cyriel Mallart, Andrew Simpkin, Rmi Venant, Nicolas Ballier, Bernardo Stearns, Jen Yu Li and Thomas Gaillat

*Japanese Lexical Complexity for Non-Native Readers: A New Dataset*

Yusuke Ide, Masato Mita, Adam Nohejl, Hiroki Ouchi and Taro Watanabe

*CEFR-based Contextual Lexical Complexity Classifier in English and French*

Desislava Aleksandrova and Vincent Pouliot

Thursday, July 13, 2023 (continued)

*The NCTE Transcripts: A Dataset of Elementary Math Classroom Transcripts*  
Dorottya Demszky and Heather Hill

*Auto-req: Automatic detection of pre-requisite dependencies between academic videos*

Rushil Thareja, Ritik Garg, Shiva Baghel, Deep Dwivedi, Mukesh Mohania and Ritvik Kulshrestha

*Transformer-based Hebrew NLP models for Short Answer Scoring in Biology*

Abigail Gurin Schleifer, Beata Beigman Klebanov, Moriah Ariely and Giora Alexandron

*Comparing Neural Question Generation Architectures for Reading Comprehension*

E. Margaret Perkoff, Abhidip Bhattacharyya, Jon Cai and Jie Cao

*A dynamic model of lexical experience for tracking of oral reading fluency*

Beata Beigman Klebanov, Michael Suhan, Zuowei Wang and Tenaha O'reilly

*Rating Short L2 Essays on the CEFR Scale with GPT-4*

Kevin P. Yancey, Geoffrey Laflair, Anthony Verardi and Jill Burstein

*Learning from Partially Annotated Data: Example-aware Creation of Gap-filling Exercises for Language Learning*

Semere Kiros Bitew, Johannes Deleu, A. Seza Doruz, Chris Develder and Thomas Demeester

*Evaluating Reading Comprehension Exercises Generated by LLMs: A Showcase of ChatGPT in Education Applications*

Changrong Xiao, Sean Xin Xu, Kunpeng Zhang, Yufang Wang and Lei Xia

*Is ChatGPT a Good Teacher Coach? Measuring Zero-Shot Performance For Scoring and Providing Actionable Insights on Classroom Instruction*

Rose Wang and Dorottya Demszky

*Does BERT Exacerbate Gender or LI Biases in Automated English Speaking Assessment?*

Alexander Kwako, Yixin Wan, Jieyu Zhao, Mark Hansen, Kai-Wei Chang and Li Cai

*MultiQG-TI: Towards Question Generation from Multi-modal Sources*

Zichao Wang and Richard Baraniuk

**Thursday, July 13, 2023 (continued)**

*Inspecting Spoken Language Understanding from Kids for Basic Math Learning at Home*

Eda Okur, Roddy Fuentes Alba, Saurav Sahay and Lama Nachman

*Socratic Questioning of Novice Debuggers: A Benchmark Dataset and Preliminary Evaluations*

Erfan Al-Hossami, Razvan Bunescu, Ryan Teehan, Laurel Powell, Khyati Mahajan and Mohsen Dorodchi

*Assessing the efficacy of large language models in generating accurate teacher responses*

Yann Hicke, Abhishek Masand, Wentao Guo and Tushaar Gangavarapu

*RETUYT-InCo at BEA 2023 Shared Task: Tuning Open-Source LLMs for Generating Teacher Responses*

Alexis Baladn, Ignacio Sastre, Luis Chiruzzo and Aiala Ros

*Empowering Conversational Agents using Semantic In-Context Learning*

Amin Omidvar and Aijun An

*NAISTeacher: A Prompt and Rerank Approach to Generating Teacher Utterances in Educational Dialogues*

Justin Vasselli, Christopher Vasselli, Adam Nohejl and Taro Watanabe

15:30 - 16:00 *Afternoon Coffee Break*

16:00 - 16:40 *Ambassador talk by Anaïs Tack (KU Leuven, imec). 'Generating Teacher Responses in Educational Dialogues: The AI Teacher Test & BEA 2023 Shared Task'*

16:40 - 17:25 *Keynote by Jordana Heller (Textio). 'Interrupting Linguistic Bias in Written Communication with NLP tools'*

17:25 - 17:30 *Closing Remarks*



# LFTK: Handcrafted Features in Computational Linguistics

Bruce W. Lee<sup>1,2,3</sup>, Jason Hyung-Jong Lee<sup>2</sup>

<sup>1</sup>University of Pennsylvania

<sup>2</sup>LXPER AI Research

brucelws@seas.upenn.edu

jasonlee@lxper.com

## Abstract

Past research has identified a rich set of handcrafted linguistic features that can potentially assist various tasks. However, their extensive number makes it difficult to effectively select and utilize existing handcrafted features. Coupled with the problem of inconsistent implementation across research works, there has been no categorization scheme or generally-accepted feature names. This creates unwanted confusion. Also, most existing handcrafted feature extraction libraries are not open-source or not actively maintained. As a result, a researcher often has to build such an extraction system from the ground up.

We collect and categorize more than 220 popular handcrafted features grounded on past literature. Then, we conduct a correlation analysis study on several task-specific datasets and report the potential use cases of each feature. Lastly, we devise a multilingual handcrafted linguistic feature extraction system in a systematically expandable manner. We open-source our system for public access to a rich set of pre-implemented handcrafted features. Our system is coined LFTK and is the largest of its kind. Find at [github.com/brucewlee/lftk](https://github.com/brucewlee/lftk).

## 1 Introduction

Handcrafted linguistic features have long been inseparable from natural language processing (NLP) research. Even though automatically-generated features (e.g., Word2Vec, BERT embeddings) have recently been mainstream focus due to fewer manual efforts required, handcrafted features (e.g., type-token ratio) are still actively found in currently literature trend (Weiss and Meurers, 2022; Campillo-Ageitos et al., 2021; Chatzipanagiotidis et al., 2021; Kamyab et al., 2021; Qin et al., 2021; Esmaeilzadeh and Taghva, 2021). Therefore, it is evident that there is a constant demand for both

<sup>3</sup>Core contributor

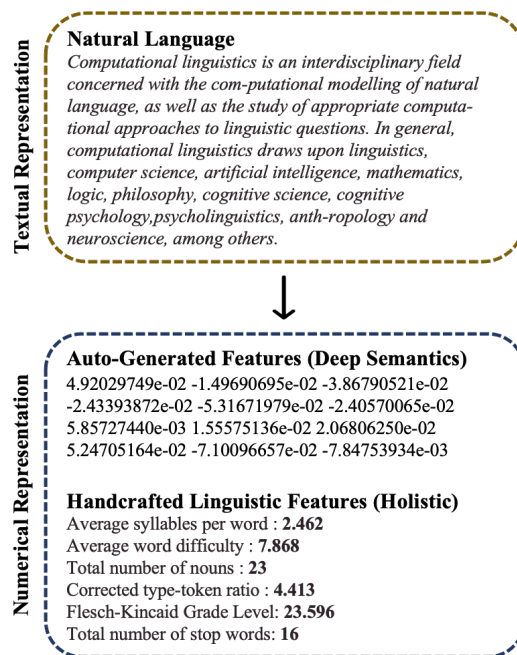


Figure 1: Difference between auto-generated (deep semantic embeddings) and handcrafted features.

the identification of new handcrafted features and utilization of existing handcrafted features.

After reviewing the recent research, we observed that most research on automatically-generated features tends to focus on creating **deeper** semantic representations of natural language. On the other hand, researchers use handcrafted features to create **wider** numerical representations, encompassing syntax, discourse, and others. An interesting new trend is that these handcrafted features are often used to assist auto-generated features in creating **wide** and **deep** representations for applications like English readability assessment (Lee et al., 2021) and automatic essay scoring (Uto et al., 2020).

The trend was observed across various tasks and languages. For example, there are Arabic speech synthesis (Amrouche et al., 2022), Burmese translation (Hlaing et al., 2022), English-French term alignment (Repar et al., 2022), German readability assessment (Blaneck et al., 2022), Italian pre-

trained language model analysis (Miaschi et al., 2020), Korean news quality prediction (Choi et al., 2021), and Spanish hate-speech detection (García-Díaz et al., 2022) systems.

Though using handcrafted features seems to benefit multiple research fields, current feature extraction practices suffer from critical weaknesses. One is the inconsistent implementations of the same handcrafted feature across research works. For example, the exact implementation of the *average words per sentence* feature can be different in Lee et al. (2021) and Pitler and Nenkova (2008) even though both works deal with text readability. Also, there have been no standards for categorizing these handcrafted features, which furthers the confusion.

In addition, no open-source feature extraction system works multilingual, though handcrafted features are increasingly used in non-English applications. The handcrafted linguistic features can be critical resources for understudied or low-resource languages because they often lack high-performance textual encoding models like BERT. In such cases, handcrafted features can be useful in creating text embeddings for machine learning studies (Zhang et al., 2022; Kruse et al., 2021; Maa-muujav et al., 2021). In this paper, we make two contributions to address the shortcomings in the current handcrafted feature extraction practices.

**1. We systematically categorize an extensive set of reported handcrafted features and create a feature extraction toolkit.** The main contribution of this paper is that we collect more than 200 handcrafted features from diverse NLP research, like text readability assessment, and categorize them. We take a systematic approach for easiness in future expansion. Notably, we designed the system so that a fixed set of *foundation features* can build up to various *derivation features*. We then categorize the implemented features into four linguistic branches and 12 linguistic families, considering the original author’s intention. The linguistic features are also labeled with available language, depending on whether our system can extract the feature in a language-agnostic manner. LFTK (**L**inguistic **F**eature **T**ool**K**it) is built on top of another open-source library, spaCy<sup>1</sup>, to ensure high-performance parsing, multilingualism, and future reproducibility by citing a specific version. Our feature extraction software aims to cover most of the generally found handcrafted linguistic features in recent research.

<sup>1</sup>[github.com/explosion/spaCy](https://github.com/explosion/spaCy)

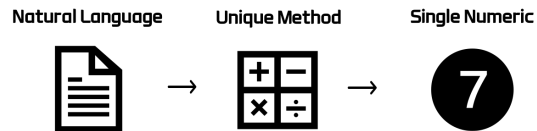


Figure 2: The three constituents of a handcrafted linguistic feature.

**2. We report basic correlation analysis on various task-specific datasets.** Due to the nature of the tasks, most handcrafted features are from text readability assessment or linguistic analysis studies with educational applications in mind. The broader applications of these handcrafted features to other fields, like text simplification or machine translation corpus generation, have been only reported fairly recently (Brunato et al., 2022; Yuksel et al., 2022). Along with the feature extraction software, we report the predictive abilities of these handcrafted features on four NLP tasks by performing a baseline correlation analysis. As we do so, we identify some interesting correlations that have not been previously reported. We believe our preliminary study can serve as a basis for future in-depth studies.

In a way, we aim to address the recent concern about the lack of ready-to-use code artifacts for handcrafted features (Vajjala, 2022). Through this work, we hope to improve the general efficiency of identifying and implementing handcrafted features for researchers in related fields.

## 2 Related Work

### 2.1 What are Handcrafted Features?

The type of linguistic feature we are interested in is often referred to as *handcrafted linguistic feature*, a term found throughout NLP research (Choudhary and Arora, 2021; Chen et al., 2021; Albadi et al., 2019; Bogdanova et al., 2017). Though the term “handcrafted linguistic features” is loosely defined, there seems to be some unspoken agreement among existing works. In this work, we define a handcrafted linguistic feature as *a single numerical value produced by a uniquely identifiable method on any natural language* (refer to Figure 2).

Unlike automatic or computer-generated linguistic features, these handcrafted features are often manually defined by combining the text’s features with simple mathematical operations like root or division (Lee et al., 2021). For example, the *average difficulty of words* (calculated with an external word difficulty-labeled database) can be considered

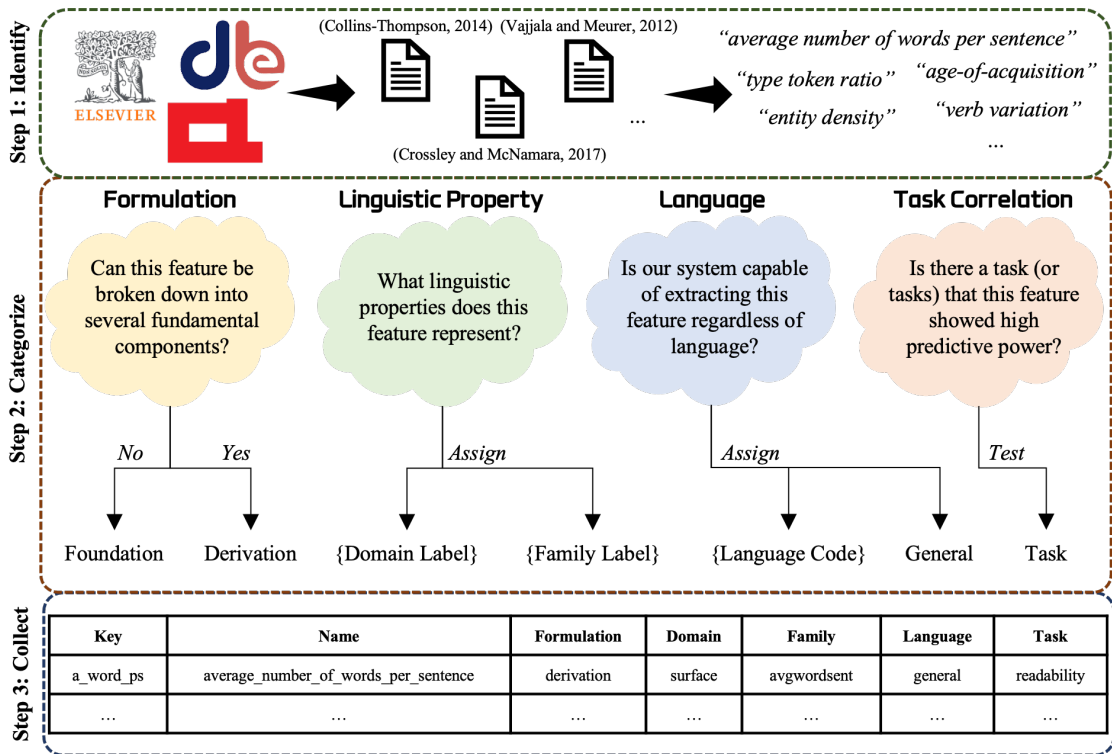


Figure 3: This diagram shows how we collected all handcrafted linguistic features implemented in our extraction software. This is also our general framework for categorizing features for future expansion too.

a handcrafted feature (Lee and Lee, 2020). Though the scope of what can be considered a single handcrafted feature is very broad, each feature always produces a single float or integer as the result of the calculation. More examples of such handcrafted features will appear as we proceed.

## 2.2 Hybridization of Handcrafted Features

It takes a great deal of effort to make automatic or computer-generated linguistic features capture the full linguistic properties of a text, other than its semantic meaning (Gong et al., 2022; Hewitt and Manning, 2019). For example, making BERT encodings capture **both** semantics and syntax with high quality can be difficult (Liu et al., 2020). On the other hand, combining handcrafted features to capture wide linguistic properties, such as syntax or discourse, can be methodically simpler. Hence, handcrafted features are often infused with neural networks in the last classification layer or directly with a sentence’s semantic embedding to enhance the model’s ability in holistic understanding (Hou et al., 2022; Lee et al., 2021). Such *feature hybridization* techniques are found in multiple NLP tasks like readability assessment (Vajjala, 2022) and essay scoring (Ramesh and Sanampudi, 2022).

## 2.3 Handcrafted Features in Recent Studies

Until recently, NLP tasks that require a holistic understanding of a given text have utilized machine learning models based only on handcrafted linguistic features. Such tasks include L2 learner’s text readability assessment (Lee and Lee, 2020), fake news detection (Choudhary and Arora, 2021), bias detection (Spinde et al., 2021), learner-based reading passage selection (Lee and Lee, 2022). Naturally, these fields have handcrafted and identified a rich set of linguistic features we aim to collect in this study. We highlight text readability assessment research as an important source of our implemented features. Such studies often involve 80~255 features from diverse linguistic branches of advanced semantics (Lee et al., 2021), discourse (Feng et al., 2010), and syntax (Xia et al., 2016).

## 3 Assembling a Large-Scale Handcrafted Linguistic Feature Extractor

### 3.1 Overview

By exploring past works that deal with handcrafted linguistic features, we aim to implement a comprehensive set of features. These features are commonly found across NLP tasks, but ready-to-use

Type	Name	Description	Example
Branch	Lexico-Semantics	attributes associated with words	Total Word Difficulty Score
Branch	Discourse	high-level dependencies between words and sentences	Total # of Named Entities
Branch	Syntax	arrangement of words and phrases	Total # of Nouns
Branch	Surface	no specifiable linguistic property	Total # of Words

Table 1: All available linguistic branches at the current version of our extraction software. The feature names in the example column are given in abbreviated formats due to space limits. We use # to indicate “number of”.

Type	Name	Description	Example
Family (F.)	WordSent	basic counts of characters, syllables, words, and sentences	Total # of Sentences
Family (F.)	WordDiff	word difficulty, frequency, and familiarity statistics	Total Word Difficulty Score
Family (F.)	PartOfSpeech	features that deal with POS (UPOS*)	Total # of Verbs
Family (F.)	Entity	named entities or entities, such as location or person	Total # of Named Entities
Family (D.)	AvgWordSent	averages of WordSent features per word, sentence, etc.	Avg. # of Words per Sentence
Family (D.)	AvgWordDiff	averages of WordDiff features per word, sentence, etc.	Avg. Word Difficulty per Word
Family (D.)	AvgPartOfSpeech	averages of PartOfSpeech features per word, sentence, etc.	Avg. # of Verbs per Sentence
Family (D.)	AvgEntity	averages of Entities features per word, sentence, etc.	Avg. # of Entities per Word
Family (D.)	LexicalVariation	features that measure lexical variation (that are not TTR)	Squared Verb Variation
Family (D.)	TypeTokenRatio	type-token ratio statistics to capture lexical richness	Corrected Type Token Ratio
Family (D.)	ReadFormula	traditional readability formulas	Flesch-Kincaid Grade Level
Family (D.)	ReadTimeFormula	basic reading time formulas	Reading Time of Fast Readers

Table 2: All available linguistic families at the current version of our extraction software. As explained in section 3.2.2, family is either *F.*: Foundation or *D.*: Derivation. \*UPOS refers to Universal POS <universaldependencies.org/u/pos/>.

public codes rarely exist. We collected and categorized over 200 handcrafted features from past research works, mostly on text readability assessment, automated essay scoring, fake news detection, and paraphrase detection. These choices of works are due to their natural intimate relationships with handcrafted features and also, admittedly, due to the authors’ limited scope of expertise. Figure 3 depicts our general process of implementing a single feature. Tables 1 and 2 show more details on categorization.

## 3.2 Categorization

### 3.2.1 Formulation

The main idea behind our system is that most handcrafted linguistic features can be broken down into multiple fundamental blocks. Depending on whether a feature can be split into smaller building blocks, we categorized all collected features into either foundation or derivation. Then, we designed the extraction system to build all derivation features on top of the corresponding foundation features. This enables us to exploit all available combinations efficiently and ensure a unified extraction algorithm across features of similar properties.

The derivation features are simple mathematical combinations of one or more foundation features. For example, the *average number of words per sen-*

*tence* is a derivation feature, defined by dividing *total number of words* by *total number of sentences*. A foundation feature can be the fundamental building block of several derivation features. But again, a foundation feature cannot be split into smaller building blocks. We build 155 derivation features out of 65 foundation features in the current version.

### 3.2.2 Linguistic Property

Each handcrafted linguistic feature represents a certain linguistic property. But it is often difficult to pinpoint the exact property because features tend to correlate with one another. Such collinear inter-dependencies have been reported by multiple pieces of literature (Imperial et al., 2022; Lee and Lee, 2020). Hence, we only categorize all features into the broad linguistic branches of lexico-semantics, syntax, discourse, and surface. The surface branch can also hold features that do not belong to any specific linguistic branch. The linguistic branches are categorized in reference to Collins-Thompson (2014). We mainly considered the original author’s intention when assigning a linguistic branch in unclear cases.

Apart from linguistic branches, handcrafted features are also categorized into linguistic families. The linguistic families are meant to group features into smaller subcategories. The main function of linguistic family is to enable efficient feature search.

		Foundation A	
		General	Specific
Foundation B	General	<i>General</i>	<i>Specific</i>
	Specific	<i>Specific</i>	<i>Specific</i>

Table 3: A theoretical example of determining the applicable language of a derivation feature that builds on top of two foundation features.

All family names are unique, and each family belongs to a specific formulation type. This means that the features in a family are either all foundation or all derivation. A linguistic family also serves as a building block of our feature extraction system. Our extraction program is a linked collection of several feature extraction modules, each representing a linguistic family (refer to Figure 4).

### 3.2.3 Applicable Language

Since handcrafted features are increasingly used for non-English languages, it is important to deduce whether a feature is generally extractable across languages. Though our extraction system is also designed with English applications in mind, we devised a systematic approach to deduce if an implemented feature is language agnostic. Like the example in Table 3, we only classify a derivation feature as generally applicable if all its components (foundation features) are generally applicable.

We can take the example of the *average number of nouns per sentence*, defined by dividing *total number of nouns* by *total number of sentences*. Since both component foundation features are generally applicable (we use UPOS tagging scheme), we can deduce that the derivation is generally applicable too. On the other hand, *Flesch-Kincaid Grade Level* (FKGL) is not generally applicable because our syllables counter is English-specific.

$$\text{FKGL} = 0.39 \cdot \frac{\# \text{ word}}{\# \text{ sent}} + 11.8 \cdot \frac{\# \text{ syllable}}{\# \text{ word}} - 15.59$$

There is no guarantee that a feature works similarly in multiple languages. The usability of a feature in a new language is subject to individual exploration.

## 3.3 Feature Details by Linguistic Family

Due to space restrictions, we only report the number of implemented features in Tables 4 and 5. A full list of these features is available in the Appendices. The following sections are used to elaborate on the motivations and implementations behind features.

Name	Feature Count
Lexico-Semantics	70
Discourse	57
Syntax	69
Surface	24
Total	220

Table 4: Feature count by branch

Name	Feature Count
WordSent	9
WordDiff	3
PartOfSpeech	34
Entity	19
AvgWordSent	7
AvgWordDiff	6
AvgPartOfSpeech	34
AvgEntity	38
LexicalVariation	51
TypeTokenRatio	10
ReadFormula	6
ReadTimeFormula	3
Total	220

Table 5: Feature count by family

### 3.3.1 WordSent & AvgWordSent

WordSent is a family of foundation features for character, syllable, word, and sentence count statistics. With the exception of syllables, this family heavily depends on spaCy for tokenization. SpaCy is a high-accuracy parser module that has been used as a base tokenizer in several multilingual projects like the Berkeley Neural Parser (Kitaev et al., 2019). We use a custom syllables count algorithm.

AvgWordSent is a family of derivation features for averaged character, syllable, word, and sentence count statistics. An example is the *average number of syllables per word*, a derivation of the *total number of words* and the *total number of syllables* foundation features.

### 3.3.2 WordDiff & AvgWordDiff

WordDiff is a family of foundation features for word difficulty analysis. This is a major topic in educational applications and second language acquisition studies, represented by age-of-acquisition (AoA, the age at which a word is learned) and corpus-based word frequency studies. Notably, there is the Kuperman AoA rating of over 30,000 words (Kuperman et al., 2012), an implemented feature in our extraction system. Another implemented feature is the word frequency statistics based on SUBLTEXus research, an improved word frequency measure based on American English sub-

titles (Brysbaert et al., 2012). AvgWordDiff averages the WordDiff features by word or sentence counts. This enables features like the *average Kuperman’s age-of-acquisition per word*.

### 3.3.3 PartOfSpeech & AvgPartOfSpeech

PartOfSpeech is a family of foundation features that count part-of-speech (POS) properties on the token level based on dependency parsing. Here, we use spaCy’s dependency parser, which is available in multiple languages. All POS counts are based on the UPOS tagging scheme to ensure multilingualism. These POS count-based features are found multiple times across second language acquisition research (Xia et al., 2016; Vajjala and Meurers, 2012). The features in AvgPartOfSpeech family are the averages of PartOfSpeech features by word or sentence counts. One example is the *average number of verbs per sentence*.

### 3.3.4 Entity & AvgEntity

Central to discourse analysis, Entity is a family of foundation features that count entities. Often used to represent the discourse characteristics of a text, these features have been famously utilized by a series of research works in readability assessment to measure the cognitive reading difficulty of texts for adults with intellectual disabilities (Feng et al., 2010, 2009). AvgEntity family are the averages of Entity features by word or sentence counts. One example is the *average number of “organization” entities per sentence*.

### 3.3.5 LexicalVariation

Second language acquisition research has identified that the variation of words in the same POS category can correlate with the lexical richness of a text (Vajjala and Meurers, 2012; Housen and Kuiken, 2009). One example of a derivative feature in this module is derived by dividing the *number of unique verbs* by the *number of verbs*, often referred to as “verb variation” in other literature. There are more derivations (“verb variation - 1, 2”) using squares or roots, which are also implemented in our system.

### 3.3.6 TypeTokenRatio

Type-token ratio, often called TTR, is another set of features found across second/child language acquisition research (Kettunen, 2014). This is perhaps one of the oldest lexical richness measures in a written/oral text (Hess et al., 1989; Richards, 1987). Though TypeTokenRatio features aim to measure similar textual characteristics

Pipeline	Time (sec)
en_core_web_sm + LFTK	12.12
en_core_web_md + LFTK	13.61
en_core_web_lg + LFTK	14.32
en_core_web_trf + LFTK	16.16

Table 6: Average time taken for extracting 220 handcrafted features from a dummy text of 1000 words. spaCy module is quite inconsistent in processing time, varying by at most 2~3 seconds.

as LexicalVariation features, we separated TTR into a separate family due to its unique prevalence.

### 3.3.7 ReadFormula

Before machine learning techniques were applied to text readability assessment, linear formulas were used to represent the readability of a text quantitatively (Solnyshkina et al., 2017). Recently, these formulas have been utilized for diverse NLP tasks like fake news classification (Choudhary and Arora, 2021) and authorship attribution (Uchendu et al., 2020). We have implemented the traditional readability formulas that are popularly used across recent works (Lee and Lee, 2023; Horbach et al., 2022; Gooding et al., 2021; Nahatame, 2021).

## 3.4 LFTK in Context

As we have explored, we tag each handcrafted linguistic feature with three attributes: domain, family, and language. These attributes assist researchers in efficiently searching for the feature they need, one of two research goals we mentioned in section 1. Instead of individually searching for handcrafted features, they can sort and extract features in terms of attributes.

Notably, our extraction system is fully implemented in the programming language Python, unlike other systems like Coh-Metrix (Graesser et al., 2004) and L2 Syntactic Complexity Analyzer (Lu, 2017). Considering the modern NLP research approaches (Mishra and Mishra, 2022; Sengupta, 2021; JUGRAN et al., 2021; Sarkar, 2019), the combination of open-source development and Python makes our extraction system more expandable and customizable in the community.

Time with spaCy model’s processing time is reported in Table 6. Excluding the spaCy model’s processing time (which is not a part of our extraction system), our system can extract 220 handcrafted features from a dummy text of 1000 words on an average of 10 seconds. This translates to about 0.01 seconds per word, and this result is ob-

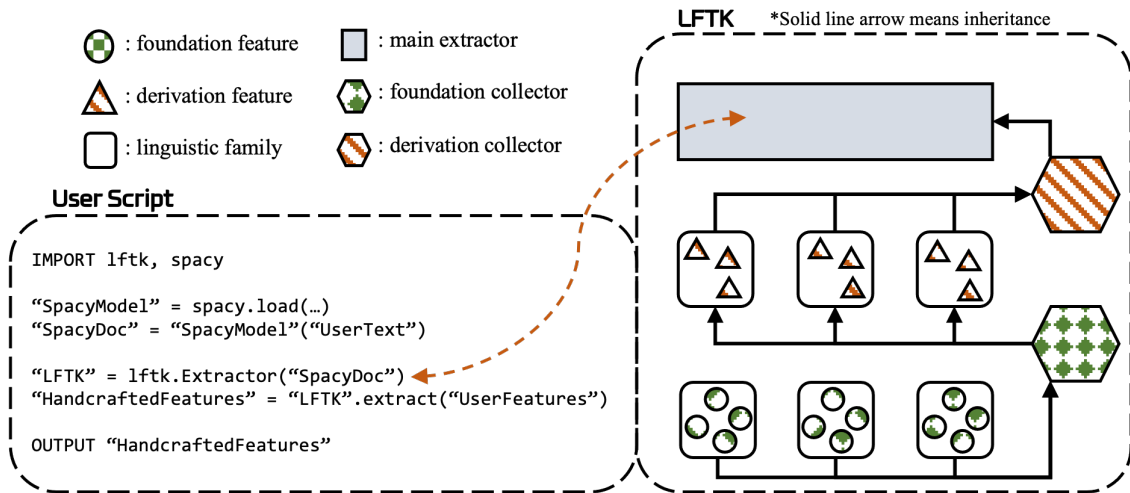


Figure 4: Schematic representation of how a user might use LFTK to extract handcrafted features. Black line arrows represent inheritance relationships. Our extraction system is a collection of multiple linguistic family modules. To interweave this program and resolve multiple dependencies, we designed a foundation collector object to inherit all foundation linguistic families first. Then all derivation linguistic families inherit the same foundation collector object. A derivation collector then inherits all derivation linguistic families, and the main extractor object inherits the derivation collector object. Considering the recent research trend, our program is solely based on the programming language Python.

tained by averaging over 20 trials of randomized dummy texts of exactly 1000 words. This time was taken with a 2.3 GHz Intel Core i9 CPU under a single-core setup. The fast extraction speed makes our extraction system suitable for large-scale corpus studies. Since our extraction system works with a wide variety of tokenizers (different accuracies and processing times) available through spaCy, one might choose an appropriate model according to the size of the studied text. Since spaCy and our extraction system are open sources registered through the Python Package Index (PyPI), reproducibility can easily be maintained by versions.

In addition, our extraction system achieves such a speed improvement due to our systematic breakdown of handcrafted features into foundation and derivation (see section 3.1.1). As depicted in Figure 4, designing the system so that derivation features are built on top of foundation features reduced duplicate program calculation to a minimum. Once a foundation feature is calculated, it is saved and used by multiple derivation features. Indeed, the *total number of words* does not have to be calculated twice for *average word difficulty per word* and *Flesch-Kincaid Grade Level*.

#### 4 Which applies to which? Task-Feature Correlation Analysis

For handcrafted features to be generally useful to the larger NLP community, it can be important to

provide researchers with a sense of which features can be potentially good in their problem setup. This section reports simple correlation analysis results of our implemented features and four NLP tasks.

To the best of our knowledge, we chose the representative dataset for each task. Table 7 reports the Pearson correlation between the feature and the dataset labels. We only report the top 10 features and bottom ten features. The full result is available in the Appendices. We used the CLEAR corpus’s *crowdsourced algorithm of reading comprehension score controlled for text length* (CAREC\_M) for readability labels on 4724 instances (Crossley et al., 2022). We used the ASAP dataset’s<sup>2</sup> *domain1\_score* on prompt 1 essays for student essay scoring labels on 1783 instances. We used the LIAR dataset for fake news labels on 10420 instances (Wang, 2017). We used SemEval 2019 Task 5 dataset’s *PS* for binary hate speech labels on 9000 instances (Basile et al., 2019).

Though limited, our preliminary correlation analysis reveals some interesting correlations that have rarely been reported. For example, *n\_verb* negatively correlates with the difficulty of a text. But there is much room to be explored. One utility behind a large-scale feature extraction system like ours is the ease of revealing novel correlations that might not have been obvious.

<sup>2</sup>[www.kaggle.com/c/asap-aes/data](http://www.kaggle.com/c/asap-aes/data)

Readability Assessment CLEAR		Essay Scoring ASAP		Fake News Detection LIAR		Hate Speech Detection SemEval-2019 Task 5	
Feature	r	Feature	r	Feature	r	Feature	r
cole	0.716	t_uword	0.832	root_num_var	0.0996	n_sym	0.134
a_char_pw	0.716	t_char	0.820	corr_num_var	0.0996	a_sym_pw	0.109
a_syll_pw	0.709	t_syll	0.819	simp_num_var	0.0992	simp_det_var	0.107
t_syll2	0.700	rt_slow	0.807	a_num_pw	0.0962	root_det_var	0.102
smog	0.685	t_word	0.807	a_num_ps	0.0855	corr_det_var	0.102
a_kup_pw	0.643	rt_fast	0.807	t_n_ent_date	0.0811	t_punct	0.097
t_syll3	0.625	rt_average	0.807	n_unum	0.0810	n_usym	0.096
fogi	0.573	t_kup	0.806	a_n_ent_date_pw	0.0772	t_sent	0.094
a_noun_pw	0.545	t_bry	0.792	a_n_ent_date_ps	0.0763	a_sym_ps	0.091
fkgl	0.544	n_noun	0.779	t_n_ent_money	0.0738	root_pron_var	0.090
...							
n_adv	-0.376	a_subtlex_us_zipf_pw	-0.295	n_uproprn	-0.0637	t_n_ent_date	-0.085
t_stopword	-0.378	simp_pron_var	-0.307	a_syll_pw	-0.0712	a_n_ent_pw	-0.086
n_uverb	-0.381	simp_part_var	-0.366	root_proprn_var	-0.0719	a_n_ent_date_pw	-0.088
simp_adp_var	-0.462	simp_aux_var	-0.399	corr_proprn_var	-0.0720	a_n_ent_gpe_pw	-0.090
a_verb_pw	-0.481	simp_cconj_var	-0.438	a_proprn_ps	-0.0745	a_adp_pw	-0.096
n_verb	-0.508	simp_ttr	-0.448	a_verb_pw	-0.0775	simp_ttr_no_lem	-0.122
n_upron	-0.531	simp_ttr_no_lem	-0.448	t_n_ent_person	-0.0790	simp_ttr	-0.122
a_pron_pw	-0.649	simp_punct_var	-0.519	a_n_ent_person_ps	-0.0822	auto	-0.156
n_pron	-0.653	simp_det_var	-0.530	a_n_ent_person_pw	-0.0850	a_char_pw	-0.167
fkre	-0.687	simp_adp_var	-0.533	a_proprn_pw	-0.0979	cole	-0.174

Table 7: Task, dataset, and top 10 correlated features (reported both in the positive and negative direction). Under our experimental setup, positive is more difficult in readability assessment. Positive is well-written in essay scoring. Positive is more truthful in fake news detection. Positive is hateful in hate speech detection. We only report feature keys due to space restrictions. The full correlation analysis and key-description pairs are available in the Appendices.

## 5 Conclusion

In this paper, we have reported our open-source, large-scale handcrafted feature extraction system. Though our extraction system covers a large set of pre-implemented features, newer, task-specific features are constantly developed. For example, *URLs count* is used for Twitter bot detection (Gilani et al., 2017) and *grammatical error count* is used for automated essay scoring (Attali and Burstein, 2006). These features, too, fall under our definition (Figure 2) of handcrafted linguistic features. Our open-source script is easily expandable, making creating a modified, research-specific version of our extraction program more convenient. With various foundation features to build from, our extraction program will be a good starting point.

Another potential user group of our extraction library is those looking to improve a neural or non-neural model’s performance by incorporating more features. Performance-wise, the breadth of linguistic coverage is often as important as selection (Lee et al., 2021; Yaneva et al., 2021; Klebanov and Madnani, 2020; Horbach et al., 2013). Our current work has various implemented features, and we believe the extraction system can be a good starting

point for many research works.

Compared to other historically important code artifacts like the Coh-Metrix (Graesser et al., 2004) and L2 Syntactic Complexity Analyzer (Lu, 2017), our extraction system is comparable or larger in size. To the best of our knowledge, this research is the first attempt to create a “general-purpose” handcrafted feature extraction system. That is, we wanted to build a system that can be widely used across NLP tasks. To do so, we have considered expandability and multilingualism from architecture design. And such consideration is grounded in the systematic categorization of popular handcrafted linguistic features into the attributes like domain and family. With the open-source release of our system, we hope that the current problems in feature extraction practices (section 1) can be alleviated.

## References

- Nuha Albadi, Maram Kurdi, and Shivakant Mishra. 2019. Investigating the effect of combining gru neural networks with handcrafted features for religious hatred detection on arabic twitter space. *Social Network Analysis and Mining*, 9(1):41.



- Aissa Amrouche, Youssouf Bentrchia, Khadidja Nesrine Boubakeur, and Ahcène Abed. 2022. Dnn-based arabic speech synthesis. In *2022 9th International Conference on Electrical and Electronics Engineering (ICEEE)*, pages 378–382. IEEE.
- Yigal Attali and Jill Burstein. 2006. Automated essay scoring with e-rater® v. 2. *The Journal of Technology, Learning and Assessment*, 4(3).
- Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th international workshop on semantic evaluation*, pages 54–63.
- Patrick Gustav Blaneck, Tobias Bornheim, Niklas Grieger, and Stephan Bialonski. 2022. [Automatic readability assessment of German sentences with transformer ensembles](#). In *Proceedings of the GemEval 2022 Workshop on Text Complexity Assessment of German Text*, pages 57–62, Potsdam, Germany. Association for Computational Linguistics.
- Dasha Bogdanova, Jennifer Foster, Daria Dziedzic, and Qun Liu. 2017. If you can't beat them join them: handcrafted features complement neural nets for non-factoid answer reranking. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 121–131.
- Dominique Brunato, Felice Dell'Orletta, and Giulia Venturi. 2022. [Linguistically-based comparison of different approaches to building corpora for text simplification: A case study on italian](#). *Frontiers in Psychology*, 13.
- Marc Brysbaert, Boris New, and Emmanuel Keuleers. 2012. Adding part-of-speech information to the sublex-us word frequencies. *Behavior research methods*, 44:991–997.
- Elena Campillo-Ageitos, Hermenegildo Fabregat, Lourdes Araujo, and Juan Martinez-Romo. 2021. Nlp-uned at erisk 2021: self-harm early risk detection with tf-idf and linguistic features. *Working Notes of CLEF*, pages 21–24.
- Savvas Chatzipanagiotidis, Maria Giagkou, and Detmar Meurers. 2021. Broad linguistic complexity analysis for greek readability classification. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 48–58.
- Mingxuan Chen, Xinqiao Chu, and KP Subbalakshmi. 2021. Mmcovar: multimodal covid-19 vaccine focused data repository for fake news detection and a baseline architecture for classification. In *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 31–38.
- Sujin Choi, Hyopil Shin, and Seung-Shik Kang. 2021. Predicting audience-rated news quality: Using survey, text mining, and neural network methods. *Digital Journalism*, 9(1):84–105.
- Anshika Choudhary and Anuja Arora. 2021. Linguistic feature based learning model for fake news detection and classification. *Expert Systems with Applications*, 169:114171.
- Kevyn Collins-Thompson. 2014. Computational assessment of text readability: A survey of current and future research. *ITL-International Journal of Applied Linguistics*, 165(2):97–135.
- Scott Crossley, Aron Heintz, Joon Suh Choi, Jordan Batchelor, Mehrnoush Karimi, and Agnes Malatinszky. 2022. A large-scaled corpus for assessing text readability. *Behavior Research Methods*, pages 1–17.
- Armin Esmaeilzadeh and Kazem Taghva. 2021. Text classification using neural network language model (nnlm) and bert: An empirical comparison. In *Proceedings of SAI Intelligent Systems Conference*, pages 175–189. Springer.
- Lijun Feng, Noémie Elhadad, and Matt Huenerfauth. 2009. Cognitively motivated features for readability assessment. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 229–237.
- Lijun Feng, Martin Jansche, Matt Huenerfauth, and Noémie Elhadad. 2010. A comparison of features for automatic readability assessment. In *Coling 2010: Posters*, pages 276–284.
- José Antonio García-Díaz, Salud María Jiménez-Zafra, Miguel Angel García-Cumbreras, and Rafael Valencia-García. 2022. Evaluating feature combination strategies for hate-speech detection in spanish using linguistic features and transformers. *Complex & Intelligent Systems*, pages 1–22.
- Zafar Gilani, Ekaterina Kochmar, and Jon Crowcroft. 2017. Classification of twitter accounts into automated agents and human users. In *Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017*, pages 489–496.
- Zheng Gong, Kun Zhou, Wayne Xin Zhao, Jing Sha, Shijin Wang, and Ji-Rong Wen. 2022. Continual pre-training of language models for math problem understanding with syntax-aware memory network. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5923–5933.
- Sian Gooding, Ekaterina Kochmar, Seid Muhie Yimam, and Chris Biemann. 2021. Word complexity is in the eye of the beholder. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4439–4449.

- Arthur C Graesser, Danielle S McNamara, Max M Louwerse, and Zhiqiang Cai. 2004. Coh-matrix: Analysis of text on cohesion and language. *Behavior research methods, instruments, & computers*, 36(2):193–202.
- Carla W Hess, Holly T Haug, and Richard G Landry. 1989. The reliability of type-token ratios for the oral language of school age children. *Journal of Speech, Language, and Hearing Research*, 32(3):536–540.
- John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138.
- Zar Zar Hlaing, Ye Kyaw Thu, Thepchai Supnithi, and Ponrudee Netisopakul. 2022. Improving neural machine translation with pos-tag features for low-resource language pairs. *Heliyon*, 8(8):e10375.
- Andrea Horbach, Alexis Palmer, and Manfred Pinkal. 2013. Using the text to evaluate short answers for reading comprehension exercises. In *Second Joint Conference on Lexical and Computational Semantics (\*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 286–295.
- Serge PJM Horbach, Jesper W Schneider, and Maxime Sainte-Marie. 2022. Ungendered writing: Writing styles are unlikely to account for gender differences in funding rates in the natural and technical sciences. *Journal of Informetrics*, 16(4):101332.
- Shudi Hou, Simin Rao, Yu Xia, and Sujian Li. 2022. Promoting pre-trained lm with linguistic features on automatic readability assessment. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, pages 430–436.
- Alex Housen and Folkert Kuiken. 2009. [Complexity, Accuracy, and Fluency in Second Language Acquisition](#). *Applied Linguistics*, 30(4):461–473.
- Joseph Marvin Imperial, Lloyd Lois Antonie Reyes, Michael Antonio Ibanez, Ranz Sapinit, and Mohammed Hussien. 2022. A baseline readability model for cebuano. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 27–32.
- SWARANJALI JUGRAN, ASHISH KUMAR, BHUPENDRA SINGH TYAGI, and VIVEK ANAND. 2021. Extractive automatic text summarization using spacy in python & nlp. In *2021 International conference on advance computing and innovative technologies in engineering (ICACITE)*, pages 582–585. IEEE.
- Marjan Kamyab, Guohua Liu, and Michael Adjeisah. 2021. Attention-based cnn and bi-lstm model based on tf-idf and glove word embedding for sentiment analysis. *Applied Sciences*, 11(23):11255.
- Kimmo Kettunen. 2014. Can type-token ratio be used to show morphological complexity of languages? *Journal of Quantitative Linguistics*, 21(3):223–245.
- Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multi-lingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505.
- Beata Beigman Klebanov and Nitin Madnani. 2020. Automated evaluation of writing—50 years and counting. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 7796–7810.
- Jessica Kruse, Paloma Toledo, Tayler B Belton, Erica J Testani, Charlesnika T Evans, William A Grobman, Emily S Miller, and Elizabeth MS Lange. 2021. Readability, content, and quality of covid-19 patient education materials from academic medical centers in the united states. *American Journal of Infection Control*, 49(6):690–693.
- Victor Kuperman, Hans Stadthagen-Gonzalez, and Marc Brysbaert. 2012. Age-of-acquisition ratings for 30,000 english words. *Behavior research methods*, 44:978–990.
- Bruce W Lee, Yoo Sung Jang, and Jason Lee. 2021. Pushing on text readability assessment: A transformer meets handcrafted linguistic features. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10669–10686.
- Bruce W Lee and Jason Lee. 2020. Lxper index 2.0: Improving text readability assessment model for 12 english students in korea. In *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 20–24.
- Bruce W Lee and Jason H Lee. 2022. Auto-select reading passages in english assessment tests? *arXiv preprint arXiv:2205.06961*.
- Bruce W Lee and Jason Hyung-Jong Lee. 2023. Traditional readability formulas compared for english. *arXiv preprint arXiv:2301.02975*.
- Tao Liu, Xin Wang, Chengguo Lv, Ranran Zhen, and Guohong Fu. 2020. Sentence matching with syntax- and semantics-aware bert. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3302–3312.
- Xiaofei Lu. 2017. Automated measurement of syntactic complexity in corpus-based l2 writing research and implications for writing assessment. *Language testing*, 34(4).

- Undarmaa Maamuujav, Carol Booth Olson, and Huy Chung. 2021. Syntactic and lexical features of adolescent 12 students' academic writing. *Journal of Second Language Writing*, 53:100822.
- Alessio Miaschi, Gabriele Sarti, Dominique Brunato, Felice Dell'Orletta, and Giulia Venturi. 2020. Italian transformers under the linguistic lens. In *CLiC-it*.
- Pradeepta Mishra and Pradeepta Mishra. 2022. Explainability for nlp. *Practical Explainable AI Using Python: Artificial Intelligence Model Explanations Using Python-based Libraries, Extensions, and Frameworks*, pages 193–227.
- Shingo Nahatame. 2021. Text readability and processing effort in second language reading: A computational and eye-tracking investigation. *Language learning*, 71(4):1004–1043.
- Emily Pitler and Ani Nenkova. 2008. Revisiting readability: A unified framework for predicting text quality. In *Proceedings of the 2008 conference on empirical methods in natural language processing*, pages 186–195.
- Han Qin, Yuanhe Tian, and Yan Song. 2021. Relation extraction with word graphs from n-grams. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2860–2868.
- Dadi Ramesh and Suresh Kumar Sanampudi. 2022. An automated essay scoring systems: a systematic literature review. *Artificial Intelligence Review*, 55(3):2495–2527.
- Andraž Repar, Senja Pollak, Matej Ulčar, and Boshko Koloski. 2022. Fusion of linguistic, neural and sentence-transformer features for improved term alignment. In *Proceedings of the BUCC Workshop within LREC 2022*, pages 61–66.
- Brian Richards. 1987. Type/token ratios: What do they really tell us? *Journal of child language*, 14(2):201–209.
- Dipanjan Sarker. 2019. *Text analytics with Python: a practitioner's guide to natural language processing*. Springer.
- Sudhriti Sengupta. 2021. Programming languages used in ai. In *Artificial Intelligence*, pages 29–35. Chapman and Hall/CRC.
- Marina Solnyshkina, Radif Zamaletdinov, Ludmila Gorodetskaya, and Azat Gabitov. 2017. Evaluating text complexity and flesch-kincaid grade level. *Journal of social studies education research*, 8(3):238–248.
- Timo Spinde, Lada Rudnitckaia, Jelena Mitrović, Felix Hamburg, Michael Granitzer, Bela Gipp, and Karsten Donnay. 2021. Automated identification of bias inducing words in news articles using linguistic and context-oriented features. *Information Processing & Management*, 58(3):102505.
- Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8384–8395.
- Masaki Uto, Yikuan Xie, and Maomi Ueno. 2020. Neural automated essay scoring incorporating hand-crafted features. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6077–6088.
- Sowmya Vajjala. 2022. Trends, limitations and open challenges in automatic readability assessment research. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5366–5377.
- Sowmya Vajjala and Detmar Meurers. 2012. On improving the accuracy of readability classification using insights from second language acquisition. In *Proceedings of the seventh workshop on building educational applications using NLP*, pages 163–173.
- William Yang Wang. 2017. “liar, liar pants on fire”: A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426.
- Zarah Weiss and Detmar Meurers. 2022. Assessing sentence readability for german language learners with broad linguistic modeling or readability formulas: When do linguistic insights make a difference? In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 141–153.
- Menglin Xia, Ekaterina Kochmar, and Ted Briscoe. 2016. Text readability assessment for second language learners. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 12–22.
- Victoria Yaneva, Daniel Jurich, Le An Ha, and Peter Baldwin. 2021. Using linguistic features to predict the response process complexity associated with answering clinical MCQs. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 223–232, Online. Association for Computational Linguistics.
- Kamer Ali Yuksel, Ahmet Gunduz, Shreyas Sharma, and Hassan Sawaf. 2022. Efficient machine translation corpus generation. In *Proceedings of the 15th biennial conference of the Association for Machine Translation in the Americas (Workshop 2: Corpus Generation and Corpus Augmentation for Machine Translation)*, pages 11–17.
- Xiaopeng Zhang, Xiaofei Lu, and Wenwen Li. 2022. Beyond differences: Assessing effects of shared linguistic features on l2 writing quality of two genres. *Applied Linguistics*, 43(1):168–195.

#	key	name	branch
1	t_word	total_number_of_words	wordsent
2	t_stopword	total_number_of_stop_words	wordsent
3	t_punct	total_number_of_punctuations	wordsent
4	t_syll	total_number_of_syllables	wordsent
5	t_syll2	total_number_of_words_more_than_two_syllables	wordsent
6	t_syll3	total_number_of_words_more_than_three_syllables	wordsent
7	t_uword	total_number_of_unique_words	wordsent
8	t_sent	total_number_of_sentences	wordsent
9	t_char	total_number_of_characters	wordsent
10	a_word_ps	average_number_of_words_per_sentence	avgwordsent
11	a_char_ps	average_number_of_characters_per_sentence	avgwordsent
12	a_char_pw	average_number_of_characters_per_word	avgwordsent
13	a_syll_ps	average_number_of_syllables_per_sentence	avgwordsent
14	a_syll_pw	average_number_of_syllables_per_word	avgwordsent
15	a_stopword_ps	average_number_of_stop_words_per_sentence	avgwordsent
16	a_stopword_pw	average_number_of_stop_words_per_word	avgwordsent
17	t_kup	total_kuperman_age_of_acquisition_of_words	worddiff
18	t_bry	total_brysaert_age_of_acquisition_of_words	worddiff
19	t_subtlex_us_zipf	total_subtlex_us_zipf_of_words	worddiff
20	a_kup_pw	average_kuperman_age_of_acquisition_of_words_per_word	avgworddiff
21	a_bry_pw	average_brysaert_age_of_acquisition_of_words_per_word	avgworddiff
22	a_kup_ps	average_kuperman_age_of_acquisition_of_words_per_sentence	avgworddiff
23	a_bry_ps	average_brysaert_age_of_acquisition_of_words_per_sentence	avgworddiff
24	a_subtlex_us_zipf_pw	average_subtlex_us_zipf_of_words_per_word	avgworddiff
25	a_subtlex_us_zipf_ps	average_subtlex_us_zipf_of_words_per_sentence	avgworddiff
26	t_n_ent	total_number_of_named_entities	entity
27	t_n_ent_person	total_number_of_named_entities_person	entity
28	t_n_ent_norp	total_number_of_named_entities_norp	entity
29	t_n_ent_fac	total_number_of_named_entities_fac	entity
30	t_n_ent_org	total_number_of_named_entities_org	entity
31	t_n_ent_gpe	total_number_of_named_entities_gpe	entity
32	t_n_ent_loc	total_number_of_named_entities_loc	entity
33	t_n_ent_product	total_number_of_named_entities_product	entity
34	t_n_ent_event	total_number_of_named_entities_event	entity
35	t_n_ent_art	total_number_of_named_entities_art	entity
36	t_n_ent_law	total_number_of_named_entities_law	entity
37	t_n_ent_language	total_number_of_named_entities_language	entity
38	t_n_ent_date	total_number_of_named_entities_date	entity
39	t_n_ent_time	total_number_of_named_entities_time	entity
40	t_n_ent_percent	total_number_of_named_entities_percent	entity

Table 8: Key, Name, and Branch. #1 ~ #40

## A All implemented features

Our extraction software is named LFTK, and its current version is **1.0.9**. Tables 8, 9, 10, and 11 reference v.1.0.9. We only report linguistic family here due to space restrictions. Though our feature description will be regularly updated at this address <sup>3</sup> whenever there is a version update, we also put the current version’s full feature table in our extraction program. Through PyPI or GitHub, the published version of our program is always retrievable.

## B Feature correlations

Tables 12, 13, 14, and 15 report the full feature correlations that are not reported in Table 7. We

have used spaCy’s en\_core\_web\_sm model, and the library version was **3.0.5**. Pearson correlation was calculated through the Pandas library, and its version was **1.1.4**. All versions reflect the most recent updates in the respective libraries.

<sup>3</sup><https://docs.google.com/spreadsheets/d/1uXtQ1ah0OL9cmHp2Hey0QcHb4bifJcQFLvYIYIAWWwQ/edit?usp=sharing>

#	key	name	branch
41	t_n_ent_money	total_number_of_named_entities_money	entity
42	t_n_ent_quantity	total_number_of_named_entities_quantity	entity
43	t_n_ent_ordinal	total_number_of_named_entities_ordinal	entity
44	t_n_ent_cardinal	total_number_of_named_entities_cardinal	entity
45	a_n_ent_pw	average_number_of_named_entities_per_word	avgentity
46	a_n_ent_person_pw	average_number_of_named_entities_person_per_word	avgentity
47	a_n_ent_norp_pw	average_number_of_named_entities_norp_per_word	avgentity
48	a_n_ent_fac_pw	average_number_of_named_entities_fac_per_word	avgentity
49	a_n_ent_org_pw	average_number_of_named_entities_org_per_word	avgentity
50	a_n_ent_gpe_pw	average_number_of_named_entities_gpe_per_word	avgentity
51	a_n_ent_loc_pw	average_number_of_named_entities_loc_per_word	avgentity
52	a_n_ent_product_pw	average_number_of_named_entities_product_per_word	avgentity
53	a_n_ent_event_pw	average_number_of_named_entities_event_per_word	avgentity
54	a_n_ent_art_pw	average_number_of_named_entities_art_per_word	avgentity
55	a_n_ent_law_pw	average_number_of_named_entities_law_per_word	avgentity
56	a_n_ent_language_pw	average_number_of_named_entities_language_per_word	avgentity
57	a_n_ent_date_pw	average_number_of_named_entities_date_per_word	avgentity
58	a_n_ent_time_pw	average_number_of_named_entities_time_per_word	avgentity
59	a_n_ent_percent_pw	average_number_of_named_entities_percent_per_word	avgentity
60	a_n_ent_money_pw	average_number_of_named_entities_money_per_word	avgentity
61	a_n_ent_quantity_pw	average_number_of_named_entities_quantity_per_word	avgentity
62	a_n_ent_ordinal_pw	average_number_of_named_entities_ordinal_per_word	avgentity
63	a_n_ent_cardinal_pw	average_number_of_named_entities_cardinal_per_word	avgentity
64	a_n_ent_ps	average_number_of_named_entities_per_sentence	avgentity
65	a_n_ent_person_ps	average_number_of_named_entities_person_per_sentence	avgentity
66	a_n_ent_norp_ps	average_number_of_named_entities_norp_per_sentence	avgentity
67	a_n_ent_fac_ps	average_number_of_named_entities_fac_per_sentence	avgentity
68	a_n_ent_org_ps	average_number_of_named_entities_org_per_sentence	avgentity
69	a_n_ent_gpe_ps	average_number_of_named_entities_gpe_per_sentence	avgentity
70	a_n_ent_loc_ps	average_number_of_named_entities_loc_per_sentence	avgentity
71	a_n_ent_product_ps	average_number_of_named_entities_product_per_sentence	avgentity
72	a_n_ent_event_ps	average_number_of_named_entities_event_per_sentence	avgentity
73	a_n_ent_art_ps	average_number_of_named_entities_art_per_sentence	avgentity
74	a_n_ent_law_ps	average_number_of_named_entities_law_per_sentence	avgentity
75	a_n_ent_language_ps	average_number_of_named_entities_language_per_sentence	avgentity
76	a_n_ent_date_ps	average_number_of_named_entities_date_per_sentence	avgentity
77	a_n_ent_time_ps	average_number_of_named_entities_time_per_sentence	avgentity
78	a_n_ent_percent_ps	average_number_of_named_entities_percent_per_sentence	avgentity
79	a_n_ent_money_ps	average_number_of_named_entities_money_per_sentence	avgentity
80	a_n_ent_quantity_ps	average_number_of_named_entities_quantity_per_sentence	avgentity
81	a_n_ent_ordinal_ps	average_number_of_named_entities_ordinal_per_sentence	avgentity
82	a_n_ent_cardinal_ps	average_number_of_named_entities_cardinal_per_sentence	avgentity
83	simp_adj_var	simple_adjectives_variation	lexicalvariation
84	simp_adp_var	simple_adpositions_variation	lexicalvariation
85	simp_adv_var	simple_adverbs_variation	lexicalvariation
86	simp_aux_var	simple_auxiliaries_variation	lexicalvariation
87	simp_cconj_var	simple_coordinating_conjunctions_variation	lexicalvariation
88	simp_det_var	simple_determiners_variation	lexicalvariation
89	simp_intj_var	simple_interjections_variation	lexicalvariation
90	simp_noun_var	simple_nouns_variation	lexicalvariation
91	simp_num_var	simple_numerals_variation	lexicalvariation
92	simp_part_var	simple_particles_variation	lexicalvariation
93	simp_pron_var	simple_pronouns_variation	lexicalvariation
94	simp_propn_var	simple_proper_nouns_variation	lexicalvariation
95	simp_punct_var	simple_punctuations_variation	lexicalvariation
96	simp_sconj_var	simple_subordinating_conjunctions_variation	lexicalvariation
97	simp_sym_var	simple_symbols_variation	lexicalvariation
98	simp_verb_var	simple_verbs_variation	lexicalvariation
99	simp_space_var	simple_spaces_variation	lexicalvariation
100	root_adj_var	root_adjectives_variation	lexicalvariation

Table 9: Key, Name, and Branch. #41 ~ #100

#	key	name	branch
101	root_adp_var	root_adpositions_variation	lexicalvariation
102	root_adv_var	root_adverbs_variation	lexicalvariation
103	root_aux_var	root_auxiliaries_variation	lexicalvariation
104	root_cconj_var	root_coordinating_conjunctions_variation	lexicalvariation
105	root_det_var	root_determiners_variation	lexicalvariation
106	root_intj_var	root_interjections_variation	lexicalvariation
107	root_noun_var	root_nouns_variation	lexicalvariation
108	root_num_var	root_numerals_variation	lexicalvariation
109	root_part_var	root_particles_variation	lexicalvariation
110	root_pron_var	root_pronouns_variation	lexicalvariation
111	root_propn_var	root_proper_nouns_variation	lexicalvariation
112	root_punct_var	root_punctuations_variation	lexicalvariation
113	root_sconj_var	root_subordinating_conjunctions_variation	lexicalvariation
114	root_sym_var	root_symbols_variation	lexicalvariation
115	root_verb_var	root_verbs_variation	lexicalvariation
116	root_space_var	root_spaces_variation	lexicalvariation
117	corr_adj_var	corrected_adjectives_variation	lexicalvariation
118	corr_adp_var	corrected_adpositions_variation	lexicalvariation
119	corr_adv_var	corrected_adverbs_variation	lexicalvariation
120	corr_aux_var	corrected_auxiliaries_variation	lexicalvariation
121	corr_cconj_var	corrected_coordinating_conjunctions_variation	lexicalvariation
122	corr_det_var	corrected_determiners_variation	lexicalvariation
123	corr_intj_var	corrected_interjections_variation	lexicalvariation
124	corr_noun_var	corrected_nouns_variation	lexicalvariation
125	corr_num_var	corrected_numerals_variation	lexicalvariation
126	corr_part_var	corrected_particles_variation	lexicalvariation
127	corr_pron_var	corrected_pronouns_variation	lexicalvariation
128	corr_propn_var	corrected_proper_nouns_variation	lexicalvariation
129	corr_punct_var	corrected_punctuations_variation	lexicalvariation
130	corr_sconj_var	corrected_subordinating_conjunctions_variation	lexicalvariation
131	corr_sym_var	corrected_symbols_variation	lexicalvariation
132	corr_verb_var	corrected_verbs_variation	lexicalvariation
133	corr_space_var	corrected_spaces_variation	lexicalvariation
134	simp_ttr	simple_type_token_ratio	typetokenratio
135	root_ttr	root_type_token_ratio	typetokenratio
136	corr_ttr	corrected_type_token_ratio	typetokenratio
137	bilog_ttr	bilogarithmic_type_token_ratio	typetokenratio
138	uber_ttr	uber_type_token_ratio	typetokenratio
139	simp_ttr_no_lem	simple_type_token_ratio_no_lemma	typetokenratio
140	root_ttr_no_lem	root_type_token_ratio_no_lemma	typetokenratio
141	corr_ttr_no_lem	corrected_type_token_ratio_no_lemma	typetokenratio
142	bilog_ttr_no_lem	bilogarithmic_type_token_ratio_no_lemma	typetokenratio
143	uber_ttr_no_lem	uber_type_token_ratio_no_lemma	typetokenratio
144	n_adj	total_number_of_adjectives	partofspeech
145	n_adp	total_number_of_adpositions	partofspeech
146	n_adv	total_number_of_adverbs	partofspeech
147	n_aux	total_number_of_auxiliaries	partofspeech
148	n_cconj	total_number_of_coordinating_conjunctions	partofspeech
149	n_det	total_number_of_determiners	partofspeech
150	n_intj	total_number_of_interjections	partofspeech
151	n_noun	total_number_of_nouns	partofspeech
152	n_num	total_number_of_numerals	partofspeech
153	n_part	total_number_of_particles	partofspeech
154	n_pron	total_number_of_pronouns	partofspeech
155	n_propn	total_number_of_proper_nouns	partofspeech
156	n_punct	total_number_of_punctuations	partofspeech
157	n_sconj	total_number_of_subordinating_conjunctions	partofspeech
158	n_sym	total_number_of_symbols	partofspeech
159	n_verb	total_number_of_verbs	partofspeech
160	n_space	total_number_of_spaces	partofspeech

Table 10: Key, Name, and Branch. #101 ~ #160

#	key	name	branch
161	n_uadj	total_number_of_unique_adjectives	partofspeech
162	n_uadp	total_number_of_unique_adpositions	partofspeech
163	n_uadv	total_number_of_unique_adverbs	partofspeech
164	n_uaux	total_number_of_unique_auxiliaries	partofspeech
165	n_ucconj	total_number_of_unique_coordinating_conjunctions	partofspeech
166	n_udet	total_number_of_unique_determiners	partofspeech
167	n_uintj	total_number_of_unique_interjections	partofspeech
168	n_unoun	total_number_of_unique_nouns	partofspeech
169	n_unum	total_number_of_unique_numerals	partofspeech
170	n_upart	total_number_of_unique_particles	partofspeech
171	n_upron	total_number_of_unique_pronouns	partofspeech
172	n_uproprn	total_number_of_unique_proper_nouns	partofspeech
173	n_upunct	total_number_of_unique_punctuations	partofspeech
174	n_usconj	total_number_of_unique_subordinating_conjunctions	partofspeech
175	n_usym	total_number_of_unique_symbols	partofspeech
176	n_uverb	total_number_of_unique_verbs	partofspeech
177	n_uspace	total_number_of_unique_spaces	partofspeech
178	a_adj_pw	average_number_of_adjectives_per_word	avgpartofspeech
179	a_adp_pw	average_number_of_adpositions_per_word	avgpartofspeech
180	a_adv_pw	average_number_of_adverbs_per_word	avgpartofspeech
181	a_aux_pw	average_number_of_auxiliaries_per_word	avgpartofspeech
182	a_cconj_pw	average_number_of_coordinating_conjunctions_per_word	avgpartofspeech
183	a_det_pw	average_number_of_determiners_per_word	avgpartofspeech
184	a_intj_pw	average_number_of_interjections_per_word	avgpartofspeech
185	a_noun_pw	average_number_of_nouns_per_word	avgpartofspeech
186	a_num_pw	average_number_of_numerals_per_word	avgpartofspeech
187	a_part_pw	average_number_of_particles_per_word	avgpartofspeech
188	a_pron_pw	average_number_of_pronouns_per_word	avgpartofspeech
189	a_proprn_pw	average_number_of_proper_nouns_per_word	avgpartofspeech
190	a_punct_pw	average_number_of_punctuations_per_word	avgpartofspeech
191	a_sconj_pw	average_number_of_subordinating_conjunctions_per_word	avgpartofspeech
192	a_sym_pw	average_number_of_symbols_per_word	avgpartofspeech
193	a_verb_pw	average_number_of_verbs_per_word	avgpartofspeech
194	a_space_pw	average_number_of_spaces_per_word	avgpartofspeech
195	a_adj_ps	average_number_of_adjectives_per_sentence	avgpartofspeech
196	a_adp_ps	average_number_of_adpositions_per_sentence	avgpartofspeech
197	a_adv_ps	average_number_of_adverbs_per_sentence	avgpartofspeech
198	a_aux_ps	average_number_of_auxiliaries_per_sentence	avgpartofspeech
199	a_cconj_ps	average_number_of_coordinating_conjunctions_per_sentence	avgpartofspeech
200	a_det_ps	average_number_of_determiners_per_sentence	avgpartofspeech
201	a_intj_ps	average_number_of_interjections_per_sentence	avgpartofspeech
202	a_noun_ps	average_number_of_nouns_per_sentence	avgpartofspeech
203	a_num_ps	average_number_of_numerals_per_sentence	avgpartofspeech
204	a_part_ps	average_number_of_particles_per_sentence	avgpartofspeech
205	a_pron_ps	average_number_of_pronouns_per_sentence	avgpartofspeech
206	a_proprn_ps	average_number_of_proper_nouns_per_sentence	avgpartofspeech
207	a_punct_ps	average_number_of_punctuations_per_sentence	avgpartofspeech
208	a_sconj_ps	average_number_of_subordinating_conjunctions_per_sentence	avgpartofspeech
209	a_sym_ps	average_number_of_symbols_per_sentence	avgpartofspeech
210	a_verb_ps	average_number_of_verbs_per_sentence	avgpartofspeech
211	a_space_ps	average_number_of_spaces_per_sentence	avgpartofspeech
212	fkre	flesch_kincaid_reading_ease	readformula
213	fkg1	flesch_kincaid_grade_level	readformula
214	fogi	gunning_fog_index	readformula
215	smog	smog_index	readformula
216	cole	coleman_liau_index	readformula
217	auto	automated_readability_index	readformula
218	rt_fast	reading_time_for_fast_readers	readtimeformula
219	rt_average	reading_time_for_average_readers	readtimeformula
220	rt_slow	reading_time_for_slow_readers	readtimeformula

Table 11: Key, Name, and Branch. #161 ~ #220

Readability Assessment CLEAR		Essay Scoring ASAP		Fake News Detection LIAR		Hate Speech Detection SemEval-2019 Task 5	
Feature	r	Feature	r	Feature	r	Feature	r
cole	0.716	t_uword	0.832	root_num_var	0.100	n_sym	0.134
a_char_pw	0.716	t_char	0.820	corr_num_var	0.100	a_sym_pw	0.109
a_syll_pw	0.709	t_syll	0.819	simp_num_var	0.099	simp_det_var	0.107
t_syll2	0.700	rt_slow	0.807	a_num_pw	0.096	root_det_var	0.102
smog	0.685	t_word	0.807	a_num_ps	0.086	corr_det_var	0.102
a_kup_pw	0.643	rt_fast	0.807	t_n_ent_date	0.081	t_punct	0.097
t_syll3	0.625	rt_average	0.807	n_unum	0.081	n_usym	0.096
fogi	0.573	t_kup	0.806	a_n_ent_date_pw	0.077	t_sent	0.094
a_noun_pw	0.545	t_bry	0.792	a_n_ent_date_ps	0.076	a_sym_ps	0.091
fkgl	0.544	n_noun	0.779	t_n_ent_money	0.074	root_pron_var	0.090
t_syll	0.527	t_subtlex_us_zipf	0.770	t_n_ent_percent	0.074	corr_pron_var	0.090
a_noun_ps	0.511	n_unoun	0.752	a_adj_ps	0.073	n_pron	0.083
auto	0.498	n_uverb	0.749	a_n_ent_money_pw	0.073	simp_pron_var	0.080
a_bry_pw	0.495	n_punct	0.740	a_n_ent_percent_pw	0.073	n_upron	0.080
a_syll_ps	0.475	t_syll2	0.739	n_adj	0.071	n_verb	0.078
n_noun	0.454	t_punct	0.738	n_uadj	0.070	rt_fast	0.078
simp_pron_var	0.443	t_stopword	0.731	a_n_ent_money_ps	0.070	t_word	0.078
t_kup	0.442	n_adp	0.727	a_n_ent_percent_ps	0.070	rt_average	0.078
a_char_ps	0.429	n_verb	0.720	n_num	0.069	rt_slow	0.078
a_kup_ps	0.421	n_uadj	0.705	root_adj_var	0.069	n_udet	0.078
a_det_ps	0.420	root_ttr	0.696	corr_adj_var	0.069	corr_aux_var	0.075
a_det_pw	0.419	root_ttr_no_lem	0.696	a_stopword_pw	0.068	root_aux_var	0.075
t_char	0.416	corr_ttr_no_lem	0.696	a_n_ent_cardinal_pw	0.066	n_uaux	0.074
a_adp_pw	0.411	corr_ttr	0.696	simp_sconj_var	0.064	n_uverb	0.073
a_adj_ps	0.403	t_sent	0.693	root_sconj_var	0.064	a_det_pw	0.073
n_unoun	0.392	n_det	0.684	corr_sconj_var	0.064	root_verb_var	0.072
a_adp_ps	0.382	n_adj	0.678	a_n_ent_cardinal_ps	0.062	corr_verb_var	0.072
a_bry_ps	0.374	n_uadv	0.675	a_sconj_pw	0.062	simp_aux_var	0.066
a_adj_pw	0.366	n_uadp	0.667	t_stopword	0.061	corr_sym_var	0.066
n_det	0.340	corr_adj_var	0.651	a_adj_pw	0.061	root_sym_var	0.066
n_adp	0.332	root_adj_var	0.651	n_usconj	0.059	n_aux	0.066
n_adj	0.309	root_adv_var	0.634	t_n_ent_cardinal	0.059	fkre	0.064
n_uadj	0.305	corr_adv_var	0.634	a_stopword_ps	0.058	t_syll3	0.064
a_word_ps	0.289	n_adv	0.634	fkre	0.058	t_subtlex_us_zipf	0.064
t_bry	0.268	root_noun_var	0.625	n_sconj	0.058	t_uword	0.062
corr_adj_var	0.261	corr_noun_var	0.625	a_sconj_ps	0.057	t_stopword	0.061
root_adj_var	0.261	root_verb_var	0.617	simp_adj_var	0.052	t_syll	0.061
root_noun_var	0.243	corr_verb_var	0.617	root_noun_var	0.051	n_adv	0.058
corr_noun_var	0.243	n_aux	0.606	corr_noun_var	0.051	n_det	0.058
a_subtlex_us_zipf_ps	0.236	t_syll3	0.575	n_adp	0.050	n_uadv	0.056
simp_verb_var	0.235	n_upron	0.574	simp_adv_var	0.049	corr_adv_var	0.054
a_n_ent_norp_ps	0.226	n_udet	0.543	corr_adv_var	0.047	root_adv_var	0.054
a_n_ent_ps	0.212	n_cconj	0.530	root_adv_var	0.047	root_noun_var	0.050
a_n_ent_org_ps	0.208	n_pron	0.491	n_noun	0.043	corr_noun_var	0.050
a_aux_ps	0.204	t_n_ent	0.487	a_adp_ps	0.043	n_noun	0.049
a_n_ent_norp_pw	0.201	n_part	0.483	t_subtlex_us_zipf	0.042	corr_ttr	0.048
t_n_ent_norp	0.196	n_uproprn	0.469	a_noun_ps	0.042	corr_ttr_no_lem	0.048
simp_adv_var	0.195	root_proprn_var	0.466	t_kup	0.042	root_ttr	0.048
a_n_ent_gpe_ps	0.191	corr_proprn_var	0.466	t_n_ent	0.042	root_ttr_no_lem	0.048
simp_ttr_no_lem	0.180	n_uaux	0.450	n_det	0.040	a_pron_pw	0.046
simp_ttr	0.180	n_upunct	0.449	n_uadv	0.040	a_pron_ps	0.044
a_stopword_ps	0.180	n_proprn	0.430	n_unoun	0.040	simp_sym_var	0.043
simp_punct_var	0.177	n_usconj	0.387	n_adv	0.039	simp_adv_var	0.042
n_udet	0.171	n_sconj	0.353	a_n_ent_ps	0.038	simp_intj_var	0.042
a_proprn_ps	0.168	t_n_ent_org	0.334	t_bry	0.038	a_det_ps	0.041
a_n_ent_cardinal_ps	0.165	smog	0.332	root_adp_var	0.038	t_n_ent_loc	0.040
a_num_ps	0.160	n_upart	0.331	corr_adp_var	0.038	root_intj_var	0.040
uber_ttr	0.154	a_punct_ps	0.328	n_uadp	0.037	corr_intj_var	0.040
uber_ttr_no_lem	0.154	t_n_ent_date	0.327	a_subtlex_us_zipf_ps	0.037	n_unoun	0.038
root_proprn_var	0.151	a_punct_pw	0.325	a_kup_ps	0.037	n_proprn	0.037

Table 12: Task, dataset, and correlated features. Part 1.



Readability Assessment CLEAR		Essay Scoring ASAP		Fake News Detection LIAR		Hate Speech Detection SemEval-2019 Task 5	
Feature	r	Feature	r	Feature	r	Feature	r
corr_propn_var	0.151	n_ucconj	0.320	corr_punct_var	0.036	a_aux_ps	0.035
bilog_ttr	0.147	n_unum	0.297	root_punct_var	0.036	n_upropn	0.035
bilog_ttr_no_lem	0.147	n_num	0.290	a_det_ps	0.036	n_uintj	0.035
simp_propn_var	0.147	corr_num_var	0.283	n_upunct	0.036	a_aux_pw	0.034
a_punct_ps	0.145	root_num_var	0.283	a_adv_ps	0.036	a_subtlex_us_zipf_pw	0.032
a_n_ent_gpe_pw	0.142	corr_pron_var	0.258	a_adv_pw	0.034	t_n_ent_product	0.031
a_n_ent_org_pw	0.140	root_pron_var	0.258	a_subtlex_us_zipf_pw	0.033	t_kup	0.030
a_n_ent_loc_ps	0.140	t_n_ent_cardinal	0.250	t_uword	0.032	root_part_var	0.029
n_upropn	0.134	a_char_pw	0.242	a_word_ps	0.031	corr_part_var	0.029
t_n_ent_gpe	0.132	cole	0.228	a_n_ent_ordinal_ps	0.031	n_upart	0.029
a_cconj_ps	0.129	t_n_ent_person	0.228	corr_ttr	0.031	t_bry	0.029
t_n_ent_org	0.127	a_syll_pw	0.223	corr_ttr_no_lem	0.031	n_punct	0.028
a_n_ent_cardinal_pw	0.115	t_n_ent_gpe	0.214	root_ttr	0.031	simp_part_var	0.027
a_n_ent_loc_pw	0.108	a_n_ent_pw	0.207	root_ttr_no_lem	0.031	n_intj	0.027
corr_sym_var	0.105	corr_sconj_var	0.205	rt_average	0.031	a_verb_pw	0.026
root_sym_var	0.105	root_sconj_var	0.205	rt_slow	0.031	n_usconj	0.026
simp_sym_var	0.104	simp_num_var	0.202	a_bry_ps	0.031	n_sconj	0.026
t_n_ent_loc	0.101	t_n_ent_time	0.191	t_word	0.031	corr_sconj_var	0.026
n_unum	0.101	a_propn_pw	0.183	rt_fast	0.031	root_sconj_var	0.026
t_n_ent_cardinal	0.099	a_n_ent_org_pw	0.166	t_n_ent_gpe	0.030	a_verb_ps	0.026
simp_cconj_var	0.099	a_n_ent_ps	0.166	a_noun_pw	0.029	a_stopword_pw	0.025
n_usym	0.098	a_n_ent_person_ps	0.164	t_n_ent_ordinal	0.028	simp_sconj_var	0.025
corr_cconj_var	0.095	a_n_ent_person_pw	0.153	n_udet	0.028	simp_cconj_var	0.024
root_cconj_var	0.095	corr_adp_var	0.146	t_punct	0.027	n_part	0.024
a_num_pw	0.093	root_adp_var	0.146	n_cconj	0.026	t_syll2	0.024
corr_ttr_no_lem	0.090	a_adv_pw	0.145	n_punct	0.026	simp_verb_var	0.024
corr_ttr	0.090	a_n_ent_org_ps	0.143	n_ucconj	0.026	t_char	0.023
root_ttr_no_lem	0.090	simp_propn_var	0.143	a_n_ent_gpe_ps	0.025	simp_adj_var	0.022
root_ttr	0.090	a_n_ent_date_pw	0.142	corr_cconj_var	0.025	t_n_ent_org	0.021
corr_num_var	0.088	a_n_ent_date_ps	0.138	root_cconj_var	0.025	a_n_ent_loc_ps	0.020
root_num_var	0.088	a_propn_ps	0.125	a_adp_pw	0.024	root_cconj_var	0.019
a_n_ent_money_pw	0.084	a_kup_pw	0.111	a_det_pw	0.024	corr_cconj_var	0.019
a_n_ent_percent_pw	0.084	a_n_ent_time_pw	0.101	a_n_ent_ordinal_pw	0.024	a_intj_ps	0.019
simp_part_var	0.083	a_n_ent_gpe_pw	0.094	root_det_var	0.024	t_n_ent_art	0.018
a_n_ent_pw	0.082	t_n_ent_quantity	0.091	corr_det_var	0.024	corr_adj_var	0.018
t_n_ent_percent	0.082	a_n_ent_cardinal_pw	0.090	simp_cconj_var	0.023	root_adj_var	0.018
t_n_ent_money	0.082	a_num_pw	0.088	a_punct_ps	0.023	a_n_ent_loc_pw	0.018
a_n_ent_percent_ps	0.081	n_uintj	0.088	a_kup_pw	0.023	a_adv_ps	0.017
a_n_ent_money_ps	0.081	n_intj	0.088	a_n_ent_pw	0.023	a_n_ent_product_pw	0.017
n_num	0.075	a_n_ent_time_ps	0.084	t_char	0.023	root_propn_var	0.015
a_n_ent_language_ps	0.073	a_adp_pw	0.082	a_cconj_ps	0.021	corr_propn_var	0.015
a_sym_ps	0.072	corr_aux_var	0.081	a_n_ent_gpe_pw	0.020	a_adv_pw	0.014
a_sym_pw	0.071	root_aux_var	0.081	t_sent	0.019	n_space	0.014
a_n_ent_event_ps	0.071	t_n_ent_percent	0.080	simp_adp_var	0.018	simp_noun_var	0.014
a_n_ent_law_pw	0.068	t_n_ent_money	0.080	simp_noun_var	0.016	n_adj	0.013
n_sym	0.068	a_n_ent_cardinal_ps	0.080	a_n_ent_quantity_pw	0.015	a_sconj_ps	0.013
a_n_ent_quantity_ps	0.068	corr_intj_var	0.077	a_char_ps	0.014	smog	0.012
a_n_ent_law_ps	0.067	root_intj_var	0.077	t_syll	0.014	n_ucconj	0.012
t_n_ent_law	0.065	a_n_ent_gpe_ps	0.075	simp_det_var	0.014	a_stopword_ps	0.012
a_n_ent_date_ps	0.064	uber_ttr	0.070	a_cconj_pw	0.014	a_sconj_pw	0.012
a_n_ent_language_pw	0.060	uber_ttr_no_lem	0.070	a_n_ent_quantity_ps	0.012	a_n_ent_product_ps	0.011
t_n_ent_language	0.058	a_det_pw	0.068	a_bry_pw	0.012	n_uadj	0.010
a_sconj_ps	0.057	a_n_ent_quantity_pw	0.068	t_n_ent_norp	0.011	t_n_ent_norp	0.008
a_n_ent_event_pw	0.057	a_n_ent_percent_pw	0.067	n_pron	0.010	a_subtlex_us_zipf_ps	0.008
a_n_ent_quantity_pw	0.056	a_n_ent_money_pw	0.067	t_n_ent_quantity	0.010	a_noun_pw	0.008
t_n_ent_quantity	0.054	a_n_ent_percent_ps	0.067	a_n_ent_loc_ps	0.009	a_n_ent_art_pw	0.007
t_n_ent_event	0.054	a_n_ent_money_ps	0.067	a_pron_ps	0.008	uber_ttr	0.007
a_verb_ps	0.052	a_n_ent_quantity_ps	0.065	a_n_ent_event_ps	0.008	uber_ttr_no_lem	0.007
t_n_ent	0.052	simp_intj_var	0.065	a_n_ent_norp_ps	0.008	t_n_ent_ordinal	0.007
a_n_ent_product_ps	0.046	a_num_ps	0.058	t_n_ent_event	0.008	t_n_ent_money	0.006

Table 13: Task, dataset, and correlated features. Part 2.

Readability Assessment CLEAR		Essay Scoring ASAP		Fake News Detection LIAR		Hate Speech Detection SemEval-2019 Task 5	
Feature	r	Feature	r	Feature	r	Feature	r
a_propn_pw	0.044	t_n_ent_loc	0.056	n_aux	0.007	t_n_ent_percent	0.006
n_ucconj	0.042	t_n_ent_product	0.049	root_pron_var	0.007	a_punct_pw	0.005
a_n_ent_ordinal_ps	0.041	t_n_ent_fac	0.048	corr_pron_var	0.007	a_noun_ps	0.005
root_punct_var	0.038	root_sym_var	0.034	a_n_ent_time_ps	0.006	n_cconj	0.003
corr_punct_var	0.038	corr_sym_var	0.034	n_upron	0.006	t_n_ent	0.003
simp_num_var	0.032	simp_sym_var	0.034	a_n_ent_loc_pw	0.005	a_n_ent_art_ps	0.001
a_n_ent_product_pw	0.031	n_usym	0.034	simp_pron_var	0.005	a_n_ent_percent_ps	0.001
t_n_ent_product	0.030	a_adj_pw	0.030	t_n_ent_loc	0.005	a_n_ent_money_ps	0.001
a_n_ent_fac_ps	0.024	root_det_var	0.028	a_n_ent_event_pw	0.005	a_word_ps	0.001
a_n_ent_art_ps	0.023	corr_det_var	0.028	t_n_ent_time	0.002	a_n_ent_ordinal_ps	-0.001
a_n_ent_fac_pw	0.019	t_n_ent_art	0.028	n_space	0.002	a_n_ent_percent_pw	-0.002
t_n_ent_fac	0.016	a_n_ent_loc_pw	0.026	a_syll_ps	0.002	a_n_ent_money_pw	-0.002
n_propn	0.015	t_n_ent_norp	0.025	a_punct_pw	0.002	a_intj_pw	-0.002
simp_space_var	0.009	n_sym	0.021	uber_ttr_no_lem	0.001	a_n_ent_law_ps	-0.005
a_n_ent_ordinal_pw	0.005	a_n_ent_product_pw	0.020	uber_ttr	0.001	n_upunct	-0.006
corr_det_var	0.001	simp_space_var	0.019	a_n_ent_time_pw	0.001	t_n_ent_law	-0.006
root_det_var	0.001	corr_space_var	0.019	simp_sym_var	0.001	a_cconj_pw	-0.007
a_n_ent_art_pw	-0.002	root_space_var	0.019	simp_aux_var	0.000	a_n_ent_fac_pw	-0.007
t_n_ent_ordinal	-0.005	t_n_ent_ordinal	0.019	a_n_ent_norp_pw	0.000	a_space_ps	-0.008
t_n_ent_art	-0.009	a_noun_pw	0.019	root_sym_var	0.000	a_n_ent_law_pw	-0.008
t_uword	-0.010	a_n_ent_loc_ps	0.017	corr_sym_var	0.000	simp_propn_var	-0.008
a_n_ent_date_pw	-0.013	a_bry_pw	0.016	a_pron_pw	-0.001	t_n_ent_fac	-0.008
a_part_ps	-0.016	n_uspace	0.015	simp_punct_var	-0.001	simp_punct_var	-0.009
a_aux_pw	-0.022	a_adv_ps	0.011	a_n_ent_language_pw	-0.002	corr_punct_var	-0.009
t_n_ent_date	-0.025	a_n_ent_fac_pw	0.010	n_usym	-0.003	root_punct_var	-0.009
a_adv_ps	-0.033	t_n_ent_event	0.008	root_aux_var	-0.003	a_space_pw	-0.009
simp_adj_var	-0.035	a_n_ent_norp_ps	0.006	corr_aux_var	-0.003	a_n_ent_quantity_ps	-0.009
a_cconj_pw	-0.054	n_space	0.004	n_sym	-0.003	t_n_ent_quantity	-0.010
simp_noun_var	-0.063	a_n_ent_product_ps	0.004	a_aux_ps	-0.003	a_n_ent_event_pw	-0.010
root_space_var	-0.072	a_n_ent_norp_pw	0.004	n_uspace	-0.003	n_uspace	-0.010
corr_space_var	-0.072	a_n_ent_event_ps	0.001	a_sym_pw	-0.003	a_n_ent_quantity_pw	-0.011
a_sconj_pw	-0.073	a_n_ent_event_pw	-0.001	t_n_ent_language	-0.004	a_n_ent_fac_ps	-0.011
n_aux	-0.081	a_space_pw	-0.001	n_uaux	-0.005	a_part_ps	-0.011
simp_sconj_var	-0.088	a_space_ps	-0.007	a_sym_ps	-0.005	a_n_ent_time_ps	-0.012
a_n_ent_time_ps	-0.091	a_n_ent_fac_ps	-0.015	t_n_ent_product	-0.005	a_n_ent_event_ps	-0.012
n_sconj	-0.096	fogi	-0.021	a_n_ent_language_ps	-0.006	simp_adp_var	-0.013
n_cconj	-0.104	a_sym_pw	-0.023	a_n_ent_product_ps	-0.007	a_punct_ps	-0.013
n_upunct	-0.115	a_sym_ps	-0.026	auto	-0.008	t_n_ent_event	-0.013
n_usconj	-0.120	a_n_ent_art_pw	-0.030	a_space_pw	-0.009	a_n_ent_ordinal_pw	-0.014
root_part_var	-0.128	fkgl	-0.032	a_n_ent_fac_pw	-0.009	a_adj_ps	-0.014
corr_part_var	-0.128	simp_adj_var	-0.033	a_n_ent_fac_ps	-0.009	a_kup_ps	-0.015
n_uadp	-0.129	auto	-0.038	simp_verb_var	-0.010	a_cconj_ps	-0.015
root_sconj_var	-0.129	a_adj_ps	-0.040	t_n_ent_fac	-0.010	a_kup_pw	-0.016
corr_sconj_var	-0.129	corr_punct_var	-0.053	root_space_var	-0.011	t_n_ent_cardinal	-0.016
a_n_ent_person_ps	-0.140	root_punct_var	-0.053	corr_space_var	-0.011	corr_space_var	-0.019
a_n_ent_time_pw	-0.145	a_n_ent_art_ps	-0.054	t_syll3	-0.011	root_space_var	-0.019
t_n_ent_time	-0.152	a_intj_pw	-0.057	a_n_ent_law_ps	-0.012	a_part_pw	-0.019
simp_det_var	-0.154	a_det_ps	-0.064	a_n_ent_art_ps	-0.012	a_adj_pw	-0.019
corr_verb_var	-0.195	a_part_pw	-0.065	a_aux_pw	-0.012	a_n_ent_time_pw	-0.021
root_verb_var	-0.195	a_adp_ps	-0.065	a_n_ent_product_pw	-0.013	root_adp_var	-0.021
n_uspace	-0.197	a_syll_ps	-0.071	n_uintj	-0.013	corr_adp_var	-0.021
root_pron_var	-0.201	a_intj_ps	-0.074	a_n_ent_law_pw	-0.013	a_syll_ps	-0.021
corr_pron_var	-0.201	fkre	-0.075	simp_intj_var	-0.013	a_bry_ps	-0.022
a_subtlex_us_zipf_pw	-0.211	a_char_ps	-0.076	root_intj_var	-0.013	a_n_ent_norp_ps	-0.022
rt_average	-0.214	root_part_var	-0.091	corr_intj_var	-0.013	t_n_ent_time	-0.022
rt_slow	-0.214	corr_part_var	-0.091	n_intj	-0.013	simp_space_var	-0.024
t_word	-0.214	a_noun_ps	-0.096	t_n_ent_art	-0.013	n_uadp	-0.025
rt_fast	-0.214	a_kup_ps	-0.096	t_n_ent_law	-0.014	a_n_ent_norp_pw	-0.031
a_intj_ps	-0.214	simp_adv_var	-0.103	t_syll2	-0.015	a_n_ent_org_ps	-0.032
simp_aux_var	-0.214	a_bry_ps	-0.110	a_space_ps	-0.016	a_n_ent_language_pw	-0.033

Table 14: Task, dataset, and correlated features. Part 3.

Readability Assessment CLEAR		Essay Scoring ASAP		Fake News Detection LIAR		Hate Speech Detection SemEval-2019 Task 5	
Feature	r	Feature	r	Feature	r	Feature	r
a_space_ps	-0.236	a_n_ent_ordinal_pw	-0.112	simp_space_var	-0.016	n_adp	-0.034
a_intj_pw	-0.245	a_word_ps	-0.115	smog	-0.017	t_n_ent_language	-0.034
n_intj	-0.247	a_n_ent_ordinal_ps	-0.118	a_n_ent_art_pw	-0.019	a_n_ent_org_pw	-0.035
a_part_pw	-0.250	a_part_ps	-0.118	a_intj_pw	-0.019	a_bry_pw	-0.035
a_n_ent_person_pw	-0.257	a_cconj_pw	-0.133	a_intj_ps	-0.022	a_n_ent_language_ps	-0.035
simp_intj_var	-0.263	bilog_ttr_no_lem	-0.144	fogi	-0.026	a_propn_ps	-0.037
corr_adv_var	-0.266	bilog_ttr	-0.144	fkgi	-0.030	a_n_ent_cardinal_ps	-0.039
root_adv_var	-0.266	simp_sconj_var	-0.149	t_n_ent_org	-0.032	t_n_ent_person	-0.040
n_uintj	-0.267	a_subtlex_us_zipf_ps	-0.157	n_verb	-0.036	t_n_ent_gpe	-0.044
t_n_ent_person	-0.269	root_cconj_var	-0.158	a_n_ent_org_ps	-0.040	a_n_ent_cardinal_pw	-0.045
a_space_pw	-0.275	corr_cconj_var	-0.158	cole	-0.040	n_num	-0.047
root_intj_var	-0.278	simp_noun_var	-0.159	root_verb_var	-0.041	simp_num_var	-0.047
corr_intj_var	-0.278	a_verb_ps	-0.162	corr_verb_var	-0.041	n_unum	-0.048
n_space	-0.283	a_stopword_ps	-0.166	simp_propn_var	-0.043	corr_num_var	-0.050
n_part	-0.284	a_aux_pw	-0.176	n_uverb	-0.044	root_num_var	-0.050
n_upart	-0.286	a_cconj_ps	-0.177	n_upart	-0.046	a_propn_pw	-0.051
a_punct_pw	-0.287	a_sconj_pw	-0.186	n_part	-0.046	fogi	-0.053
a_stopword_pw	-0.288	a_aux_ps	-0.192	a_verb_ps	-0.047	fkgi	-0.055
t_punct	-0.290	a_pron_ps	-0.201	corr_part_var	-0.049	a_n_ent_person_pw	-0.058
n_uaux	-0.292	a_sconj_ps	-0.203	root_part_var	-0.049	a_char_ps	-0.061
n_punct	-0.301	simp_verb_var	-0.204	simp_part_var	-0.050	a_n_ent_ps	-0.062
corr_aux_var	-0.308	a_pron_pw	-0.209	a_n_ent_org_pw	-0.051	a_n_ent_person_ps	-0.062
root_aux_var	-0.308	a_verb_pw	-0.220	a_part_ps	-0.052	a_syll_pw	-0.066
a_pron_ps	-0.319	a_stopword_pw	-0.236	a_char_pw	-0.055	a_num_ps	-0.070
n_uadv	-0.333	a_subtlex_us_zipf_pw	-0.295	n_propn	-0.057	a_adp_ps	-0.073
t_subtlex_us_zipf	-0.334	simp_pron_var	-0.307	bilog_ttr_no_lem	-0.059	a_n_ent_date_ps	-0.074
a_adv_pw	-0.338	simp_part_var	-0.366	bilog_ttr	-0.059	a_n_ent_gpe_ps	-0.074
t_sent	-0.339	simp_aux_var	-0.399	simp_ttr	-0.059	a_num_pw	-0.080
corr_adp_var	-0.359	simp_cconj_var	-0.438	simp_ttr_no_lem	-0.059	bilog_ttr_no_lem	-0.083
root_adp_var	-0.359	simp_ttr	-0.448	a_part_pw	-0.060	bilog_ttr	-0.083
n_adv	-0.376	simp_ttr_no_lem	-0.448	n_upropn	-0.064	t_n_ent_date	-0.085
t_stopword	-0.378	simp_punct_var	-0.519	a_syll_pw	-0.071	a_n_ent_pw	-0.086
n_uverb	-0.381	simp_det_var	-0.530	root_propn_var	-0.072	a_n_ent_date_pw	-0.088
simp_adp_var	-0.462	simp_adp_var	-0.533	corr_propn_var	-0.072	a_n_ent_gpe_pw	-0.090
a_verb_pw	-0.481			a_propn_ps	-0.074	a_adp_pw	-0.096
n_verb	-0.508			a_verb_pw	-0.077	simp_ttr_no_lem	-0.122
n_upron	-0.531			t_n_ent_person	-0.079	simp_ttr	-0.122
a_pron_pw	-0.649			a_n_ent_person_ps	-0.082	auto	-0.156
n_pron	-0.653			a_n_ent_person_pw	-0.085	a_char_pw	-0.167
fkre	-0.687			a_propn_pw	-0.098	cole	-0.174

Table 15: Task, dataset, and correlated features. Part 4.

# Improving Mathematics Tutoring With A Code Scratchpad

**Shriyash Upadhyay**  
Martian  
yash@withmartian.com

**Etan Ginsberg**  
Martian  
etan@withmartian.com

**Chris Callison-Burch**  
University of Pennsylvania  
ccb@upenn.edu

## Abstract

Large language models can solve reasoning tasks (like math problems) more effectively when they are allowed to generate rationales. However, a good tutoring system should not just generate solutions, but should also generate explanations and should be able to correct and guide students. We show that providing a code scratchpad improves performance on each tutoring step with a gradeschool mathematics dataset. On these tutoring tasks, GPT-3 models provided with a code scratchpad significantly outperform those given only a language scratchpad (77.7% vs 48.7% cumulative accuracy).

## 1 Introduction

Intelligent Tutoring Systems (ITS) are known to be effective aids to learning, but are currently difficult and time consuming to create. Such systems can aid learning significantly despite limitations, improving student performance with a median improvement of 0.66 standard deviations (Kulik and Fletcher, 2016). However, many notable ITS (for example Chaudhri et al., 2013) have been limited due to the time-intensive and costly processes required to create them. Previous work on ITS has typically focused on rule-based methods. To the degree that large language models (LLMs) are used, it has been to generate additional rules for such systems. Recently, advances in natural language processing have pointed at the possibility of using LLMs as tutoring systems, most notably 1) the success of large language models in math world problem solving due to rationale generation (Rajani et al., 2019; Nye et al., 2021; Wei et al., 2022) and 2) the improved alignment of dialogue agents such as ChatGPT and Sparrow (Glaese et al., 2022). We conduct a feasibility study on the application of LLMs to tutoring in the context of mathematics at an elementary school level by investigating their performance on the tasks required by an ITS (see Figure 1).

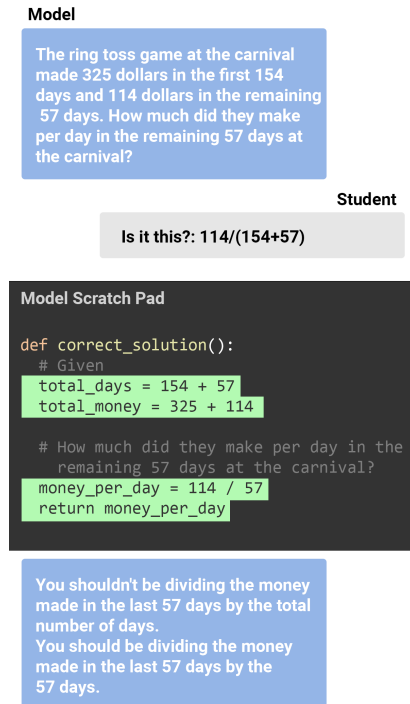


Figure 1: We evaluate the performance of two GPT-3 models on the sub-tasks present in an intelligent tutoring system, providing one with a text-only scratchpad and the other with a code scratchpad.

Our contributions are the following:

- We evaluate LLMs on the tasks present in an ITS by proving a mapping between the sub-tasks in an ITS and tasks which can be done by an LLM. Using this, we show that GPT-3 with a text-only scratchpad has a significant error rate when acting as a domain model and tutoring model.
- We show that using a code scratchpad instead of text-only ameliorates the errors in acting as a tutoring model. Combined with improved ability to solve math problems, this means GPT-3 makes a significantly better tutor with a code scratchpad (77.7% vs 48.7% cumulative accuracy on ITS sub-tasks).

## 2 Related Work & Background

Early uses of NLP in ITS involved the use of knowledge-based and rule-based systems (Hartley and Sleeman, 1973). Such systems have shown to be pedagogically effective (Kulik and Fletcher, 2016), and as such they continue to constitute the majority of ITS today. Teaching and interacting with the student in an ITS takes place through some fixed set of interactions, often mediated by extracting keywords from user utterances or as goal-oriented dialogue systems. This tends to be the case in both knowledge-based ITS (Piramuthu, 2005; Chaudhri et al., 2013), and in rule-based systems (Jarvis et al., 2004; Stamper, 2006). For open-ended domains, Named Entity Recognition (NER) has been used to determine whether a student’s open-ended response meets a set of constraints (Dzikovska et al., 2007). Techniques from NLP have also been used more selectively to implement features in these systems, such as machine translation for language learning (Moghrabi, 1998) and Automatic Speech Recognition (ASR) for audio-based tutors (Ward et al., 2011; Pradhan et al., 2016).

However, newer techniques such as LLMs have not found extensive use in implementing tutoring systems. This is despite the success of generative models such as GPT-3 (Brown et al., 2020) and PALM (Chowdhery et al., 2022) across a wide variety of tasks, the improvement in dialogue systems stemming from alignment as seen in models like ChatGPT and Sparrow (Glaese et al., 2022), and the success of LLMs (especially those that generate code) in the related domain of Math Word Problem Solving (Li et al., 2022; Gao et al., 2022). Much of the work on LLMs in education has focused on question generation as opposed to intelligent tutoring systems, for example (Dugan et al., 2022) for flashcard generation or (Sarsa et al., 2022) for programming exercises.

This may be the result of the difficulty in evaluating the quality of generations from LLMs, especially explanations for the answers that they give, as noted in (Lewkowycz et al., 2022). In this paper, we evaluate the ability of LLMs to serve as tutors, focusing on the evaluation of generated explanations and corrections.

## 3 Methodology

**Intelligent Tutoring System.** In order to evaluate the suitability of large language and code models

to tutoring, we test how well those models do in the sub-tasks typically present in Intelligent Tutoring Systems.

Intelligent tutoring systems are typically composed of four components (Nkambou et al., 2010): the domain model, student model, tutoring model, and user interface model. The domain model consists of the actions and correct steps required to solve a problem. For example, in an ITS for mathematics the domain model might consist of all the relevant operations and the correction method of solving problems. The student model consists of the actions taken by the student (for example, the scratchpad the student is using to do their work). When the student deviates from the domain model, the tutoring model provides feedback (for example, telling a student what step they should take next or what a student did wrong in their scratchpad). Finally, the user interface model facilitates interaction between the user and the tutoring model (this might be the system which parses the scratchpad and then parlays feedback to the student).

We can instantiate a tutor using an LLM by creating each of the following parts. The user interface model is simply natural language. The domain model consists of problems with correct solutions (generated by the model), the student model consists of the language produced by the student, and the tutoring model consists of comparing domain and student models in text and producing feedback. We illustrate each of the parts of an ITS and how they can be performed by an LLM in Figure 4.

**Dataset.** Following previous work, we report our results on SVAMP (Patel et al., 2021). SVAMP is a challenge dataset consisting of 1000 math word problems designed to demonstrate the failures modes of word problem solving models. The dataset focuses on arithmetic word problems, i.e. those whose solutions are a combination of numerical values and the basic arithmetic operations (+, −, ×, ÷). Examples of such problems can be found in Table 1. Each problem has both a body (containing the narrative that furnishes the relevant values and relationships) and the question being asked about that narrative. Each problem is also annotated with additional data, such as the correct numerical solution. The dataset also contains three types of "difficult" problems: problems with re-used values, problems with

---

Dave had 24 files and 13 apps on his phone. After deleting some apps and files he had 17 apps and 21 files left. How many files did he delete?

---

The grasshopper and the frog had a jumping contest. The grasshopper jumped 9 inches and the frog jumped 12 inches. How much farther did the frog jump than the grasshopper?

---

At the zoo, a cage had 95 snakes and 61 alligators. If 64 snakes were hiding How many snakes were not hiding?

---

Table 1: Examples of problems from the SVAMP dataset (Patel et al., 2021).

multiple operations, and problems with unused values.

**Models.** The large language model used in our experiments is GPT-3 (Brown et al., 2020). All experiments are run using the largest version of these models (the text scratchpad is generated with text-davinci-002 and the code scratchpad with code-davinci-002). For both models, decoding was done with nucleus sampling using  $p=1$  (Holtzman et al., 2020). The temperature parameter was 0 and the frequency penalty was 0.5. The prompts used with each model can be found in Appendix A.

**Scratchpads.** Previous work has shown that providing models with a scratchpad where they can generate rationales for their answers improves their accuracy on reasoning tasks such as math word problem solving (Rajani et al., 2019; Nye et al., 2021; Wei et al., 2022). In our work, the scratchpads are a "thinking space" for models, which would not be shown to the students, but are used to compute answers or analyze student responses.

Scratchpads can take the form of text, code, or a combination of both. When the scratchpad is purely code, we extract an answer by running the code. When the scratchpad is text or a combination of both, the model produces an answer in the form of text.

**Generating and Running Code.** All code snippets generated in this paper’s experiments are generated in the python programming language. If GPT-3 is used to generate runnable output, we generate GPT-3’s response in a function named `solution`. Any code generated outside the `solution` function is not run. In order to prevent

	Code	Text
Solved	79.4%	63.7%
Explained	98.9%	97.9%
Corrected	99.0%	78.1%
Cummulative	77.7%	48.7%

Table 2: Performance of GPT-3 with text/code scratchpads on each tutoring sub-task. The cumulative performance is the product of the performance on each sub-task.

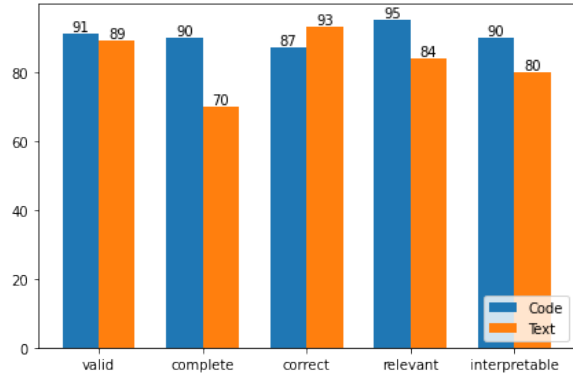


Figure 2: Results of our human evaluation for explanation generation. Numbers represent the percentage of annotations which provided a yes answer to each evaluation criterion.

multiple solution functions from being generated, we stop generation whenever GPT-3 tries to open a multi-line comment using triple quotes (`"""`).

## 4 Experiments

Our first experiment evaluates the difference in performance between text and code scratchpads in math problem solving. We evaluate, as is typical for math word problem solving, by measuring the percentage of numerically correct answers produced by the model. This is a necessary, but not sufficient, part of generating the domain model. The LLM should produce not only a correct answer, but should also provide a correct explanation to produce that answer. Therefore, our second experiment evaluates whether the model provides an acceptable explanation for its answer. Because we generate answers with GPT-3 by using CoT prompting, an explanation is automatically produced. For the code scratchpad, we generate an explanation by asking the model to convert the code used to produce an answer into plain English. These two experiments evaluate the ability of the LLMs to serve as a domain model.

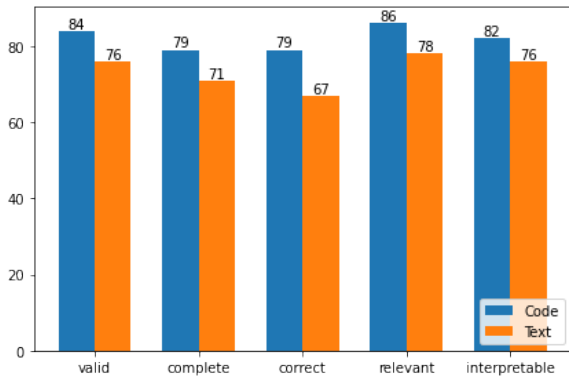


Figure 3: Results of our human evaluation for correction generation. Numbers represent the percentage of annotations which provided a yes answer to each evaluation criterion.

Our third experiment evaluates the ability of the LLMs to serve as tutoring models. We start with the correct answers and explanations provided by the model. For each question answered correctly, we prompt the models using poorly formed prompts in order to generate plausible incorrect answers (i.e. using the model to simulate the output of a student). Then, we provide the model with the incorrect answer and the correct answer, and prompt it to explain why the incorrect answer is wrong and to accordingly provide feedback to the student.

The first experiment is evaluated automatically, while the second and third experiments are evaluated by human annotators.

## 5 Evaluation

We tasked 208 annotators to evaluate the quality of explanations and corrections. Each annotator was shown 20 examples of explanations and later shown 20 examples of corrections. A total of 213 explanations and 190 corrections were evaluated in this way. We modify the question evaluation procedure in (Dugan et al., 2022) for evaluating explanations and asked the following yes/no questions:

1. (Valid) Does the explanation contain instructions which could be used to correctly answer the problem? It may also have other steps which are irrelevant or incorrect.
2. (Complete) Does the explanation explain all steps required to do the problem? That means the explanation is not missing any key steps a learner would need in order to solve such a problem.

3. (Correct) Does the explanation \*not\* contain any incorrect steps or incorrect explanation?
4. (Relevant) Does the explanation \*not\* contain information irrelevant to the problem.
5. (Interpretable) Would a student who is learning material at the level of this problem be able to understand the explanation?

If an annotator answered yes to all of the above questions, the explanation/correction was considered "acceptable"; otherwise, it was considered "unacceptable". Using Fleiss'  $\kappa$ , we observe moderate inter-annotator agreement ( $\kappa = 0.21$ ).

In Table 2 we report the overall performance with each type of scratchpad on each sub-task. Code generation outperforms text generation on all sub-tasks.

In Figure 2 we report the detailed results of our evaluation for explanations. We can see that language and code scratchpads achieve similar performance in generating explanations. This is notable because of the difference in how the two models can create explanations. Text generation, by virtue of generating a Chain of Thought, comes with an explanation. Code generation requires an additional step of transforming code into text, which introduces an opportunity for more errors. This is reflected in the fact that explanations generated in text are more likely to be correct. However, code generation is much more likely to result in a complete explanation. This makes sense, as the model must explicitly list steps in code in order for the code to compile, while text is more prone to logical leaps or implicit steps.

In Figure 3 we report the detailed results of our evaluation for corrections. In contrast with explanation generation, when generating corrections, code scratchpads encounter fewer errors of all kinds than text ones.

## 6 Conclusion & Future Work

In this work we show that large language models can perform the tasks associated with traditional Intelligent Tutoring Systems (ITS). We show that models which use text scratchpads suffer from substantial errors in solving and correcting mathematical questions, and that these errors can be ameliorated through the use of code scratchpads. Nonetheless, code generation (while accurate enough to potentially useful as tool for authoring ITS) still suffers from significant errors.

Future work should seek to further explore the applicability of LLMs to tutoring. This includes developing both new evaluation methods and new methods of reducing errors.

## 7 Limitations

**Testing Necessary, But Not Sufficient Conditions For Tutoring With LLMs.** In this paper, we test the abilities of LLMs to perform the functions present in Intelligent tutoring systems, namely generating explanations and corrections. There are also other desirable properties, like the ability to answer direct questions from a student or the ability to present content engagingly, which are beyond the scope of this paper. Indeed, those properties are some of the areas where LLMs probably excel relative to traditional ITS. We have only explored a necessary condition – are models able to reliably teach – not a sufficient set of conditions for the evaluation of tutoring using an LLM.

**Focusing On Mathematics.** In this paper, we focus on tutoring in rudimentary mathematics. While this is useful – it is a necessary condition for a useful tutoring system, especially because arithmetic skills are used in almost all domains of learning – there are many other domains to which we might want to apply tutoring. LLMs may have greater or lesser aptitude in these domains than in arithmetic. Evaluation at the level of gradeschool mathematics tells us that these models are still error prone, but does not necessarily tell us how close they are to usefulness in tutoring other subjects (either more advanced mathematics or orthogonal subjects like history or writing).

**Generalizing Text vs Code Results.** We aim to examine the differences in ability of code scratchpads and text scratchpads for the purposes of tutoring. While this paper provides evidence in that direction, we only compare two GPT-3 models: text-davinci-002 and code-davinci-002. The amount of manual effort required to evaluate explanations and correction limited the number of comparisons we could conduct, as did the limited number of highly performant code/text generating models.

## 8 Ethics Statement

By offering a highly scalable and low-cost tutoring solution, ITS offer lower income and minority

communities a critical resource in boosting educational outcomes that has historically only been available to wealthy students in the form of expensive individual private tutors. We hope that these advancements will reduce key educational disparities. It is also important in that vein to ensure that public schools with smaller budgets are given access to ITS systems in pilot trials. Instructors and students should become well-versed in using the technology in order to ensure successful expansion into such schools. Furthermore, advancements in model distillation and the creation of smaller language models will lead to lower costs for adoption for the schools that are most in need. Intelligent Tutoring Systems that run on generative AI models bring many of the same dangers of bias that are prevalent in models more generally. Gender and racial stereotypes can be invoked when students are presented with specific explanations. For example, a model may explain a math question that involved individuals choosing jobs through a hypothetical example that invokes a gender or racial stereotype based on the example given. However, recent advancements in alignment have made great strides in reducing this issue.

As these models become more widely available to students, there is an increased likelihood of students using these models for cheating on assignments that are supposed to be completed without outside resources. Unlike traditional plagiarism which can be checked by comparing document similarity, the use of generative AI to answer questions on exams and assignments is far more difficult to detect.

Lastly, discrepancies in model outputs and inaccurate answers given when some students use the ITS but not others can lead to misunderstandings and confusion amongst students. As a result, instructors should supervise the outputs given by the ITS to students. In the event that a student was supplied incorrect information by an ITS, that should be taken into account in grading that student’s course material. Instructors should incorporate AI policies in their syllabi that outline acceptable uses of ITS systems, address the handling of potential inaccuracies from those systems, and ensure all students have access to the ITS systems.

By highlighting the limitations of large language models as tutoring systems, we hope our work will prevent the premature use of these technologies.



## 9 Acknowledgements

This research is based upon work supported in part by the the NSF (Award 1928631). Approved for Public Release, Distribution Unlimited. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the NSF or the U.S. Government.

## 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, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff 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. *ArXiv*, abs/2005.14165.
- Vinay K. Chaudhri, Britte Haugan Cheng, Adam Overholtzer, Jeremy Roschelle, Aaron Spaulding, Peter Clark, Mark T. Greaves, and David Gunning. 2013. Inquire biology: A textbook that answers questions. *AI Mag.*, 34:55–72.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek B Rao, Parker Barnes, Yi Tay, Noam M. Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Benton C. Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier García, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayanan Pilla, Marie Pellat, Aitor Lewkowycz, Erica Oliveira Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Díaz, Orhan Firat, Michele Catasta, Jason Wei, Kathleen S. Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *ArXiv*, abs/2204.02311.
- Liam Dugan, Eleni Miltsakaki, Shriyash Upadhyay, Etan Ginsberg, Hannah Gonzalez, DaHyeon Choi, Chuning Yuan, and Chris Callison-Burch. 2022. [A feasibility study of answer-agnostic question generation for education](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1919–1926, Dublin, Ireland. Association for Computational Linguistics.
- Myroslava O. Dzikovska, Charles B. Callaway, Elaine Farrow, Manuel Marques-Pita, Colin Matheson, and Johanna D. Moore. 2007. Adaptive tutorial dialogue systems using deep nlp techniques. In *North American Chapter of the Association for Computational Linguistics*.
- Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2022. Pal: Program-aided language models. *ArXiv*, abs/2211.10435.
- Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, Lucy Campbell-Gillingham, Jonathan Uesato, Po-Sen Huang, Ramona Comanescu, Fan Yang, Abigail See, Sumanth Dathathri, Rory Greig, Charlie Chen, Doug Fritz, Jaume Sanchez Elias, Richard Green, Soňa Mokrá, Nicholas Fernando, Boxi Wu, Rachel Foley, Susannah Young, Iason Gabriel, William Isaac, John Mellor, Demis Hassabis, Koray Kavukcuoglu, Lisa Anne Hendricks, and Geoffrey Irving. 2022. [Improving alignment of dialogue agents via targeted human judgements](#).
- J. R. Hartley and Derek H. Sleeman. 1973. Towards more intelligent teaching systems. *International Journal of Human-computer Studies International Journal of Man-machine Studies*, 5:215–236.
- Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. *ArXiv*, abs/1904.09751.
- Matthew P. Jarvis, Goss Nuzzo-Jones, and Neil T. Hefernan. 2004. Applying machine learning techniques to rule generation in intelligent tutoring systems. In *International Conference on Intelligent Tutoring Systems*.
- James A. Kulik and John Dexter Fletcher. 2016. Effectiveness of intelligent tutoring systems. *Review of Educational Research*, 86:42 – 78.
- Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Venkatesh Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. *ArXiv*, abs/2206.14858.
- Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2022. On the advance of making language models better reasoners. *ArXiv*, abs/2206.02336.
- Chadia Moghrabi. 1998. Using language resources in an intelligent tutoring system for french. In *ACL*.

- Roger Nkambou, Jacqueline Bourdeau, and Riichiro Mizoguchi. 2010. *Advances in Intelligent Tutoring Systems*. Springer Berlin, Heidelberg.
- Maxwell Nye, Anders Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. 2021. Show your work: Scratchpads for intermediate computation with language models. *ArXiv*, abs/2112.00114.
- Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.
- Selwyn Piramuthu. 2005. Knowledge-based web-enabled agents and intelligent tutoring systems. *IEEE Transactions on Education*, 48:750–756.
- Sameer Pradhan, Ronald A. Cole, and Wayne H. Ward. 2016. My science tutor—learning science with a conversational virtual tutor. In *ACL*.
- Nazneen Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Annual Meeting of the Association for Computational Linguistics*.
- Sami Sarsa, Paul Denny, Arto Hellas, and Juho Leinonen. 2022. Automatic generation of programming exercises and code explanations using large language models. *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 1*.
- John C. Stamper. 2006. Automating the generation of production rules for intelligent tutoring systems.
- Wayne H. Ward, Ronald A. Cole, Daniel Bolaños, Cindy Buchenroth-Martin, Edward Svirsky, Sarel van Vuuren, Timothy J. Weston, Jing Zheng, and Lee Becker. 2011. My science tutor: A conversational multimedia virtual tutor for elementary school science. *ACM Trans. Speech Lang. Process.*, 7:18:1–18:29.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

## A Prompts

### A.1 Prompts Used For Math Problem Solving

#### Solving Math Problems With GPT-3

```
1 {problem.body} {problem.question}
2
3 A: Lets think step by step.
4 {model output}
5
6 So, the answer (in arabic numerals)
   is:
7 {model output}
```

#### Solving Math Problems With code

```
1 """
2 {problem.body} {problem.question}
3 """
4 {model output}
5
6 # So the answer (in arabic numerals)
   is: {model output}
```

### A.2 Prompts Used For Explanation Generation

#### Converting Code Answers To English Explanations

```
1 """
2 Write a function which computes and
   returns the solution to the
   following word problem:
3 At the zoo, a cage had 95 snakes and
   61 alligators. If 64 snakes were
   hiding How many snakes were not
   hiding?
4 The function must return a single
   numerical value. It cannot print
   the answer.
5 """
6 def solution():
7     # Given
8     snakes = 95
9     alligators = 61
10    hiding_snakes = 64
11
12
13    # How many snakes were not hiding?
14    return snakes - hiding_snakes
15
16 """
17 Here's what the above code is doing:
18 1. The problem is asking how many
   snakes were not hiding. So, we
   need to find how many snakes were
   hiding and subtract it from how
   many snakes there were. (snakes -
   hiding_snakes)
19 2. The problem tells us that there
   were 95 snakes. (snakes = 95)
20 3. The problem tells us that 64
   snakes were hiding. (
   hiding_snakes = 64)
21 4. So, the answer is 95 - 64 = 31.
22 """
23
24 {answer}
25
26 """
27 Here's what the above code is doing:
28 1. {model output}
```

### A.3 Prompts Used To Generate Incorrect Answers

#### Generating example scratchpads using Code

```
1 """
2 {problem.body} {problem.question}
3 """
4 def solution():
5     return {model output}
```

### A.4 Prompts Used For Correction Generation

#### Correcting Solutions (used for both text and code)

```
1 {problem.body} {problem.question}
2 {correct_explanation}
3 {incorrect_answer}
4
5 What approach does the correct
   solution take:
6 {model outout}
7
8 What approach does the incorrect
   solution take:
9 {model output}
10
11 Why is the incorrect solution
   incorrect:
12 {model output}
```

## B Annotation Interface

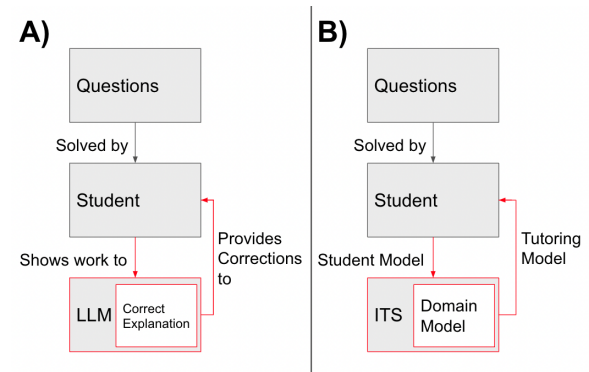


Figure 4: We evaluate the the performance of LLMs with text scratchpads and code scratchpads models in tutoring. (A) shows the parts of our system: given a question, a student produces an answer and the answer is shown to an LLM. The LLM first generates a solution to the question and a explanation for the solution. If the student gets the question wrong, the model also provides a correction. (B) shows how each of those steps corresponds to the parts of a traditional Intelligent Tutoring System (ITS). The red portion is what we evaluate: namely, the ability of the model to serve as a domain model (explanation) and tutoring model (correction).

## How To Rate Explanations

A good explanation should have the following properties:

**Contains a valid solution** The explanation contains instructions which could be used to correctly answer the problem. (It may also have other steps which are irrelevant or incorrect.)

**Is complete** The explanation explains all steps required to do the problem. That means the explanation is not missing any key steps a learner would need in order to solve such a problem.

**Is correct** The explanation doesn't contain an incorrect step or incorrect explanation.

**Is relevant** The explanation doesn't contain information irrelevant to the problem.

**Is interpretable** A student who is learning material at the level of this problem would be able to understand the explanation.

You will check boxes indicating whether the explanation that the model gave has these properties. You can also provide additional comments on the explanations the model produces via a text form.

Figure 5: The annotation guidelines shown to annotators immediately before annotation. Complementary instructions were given prior to annotation.

The Question:	The Explanation:	Does the explanation meet the criteria?
Jake has 18 fewer peaches than Steven who has 13 more peaches than Jill. Steven has 19 peaches. How many peaches does Jill have?	<ol style="list-style-type: none"><li>1. The problem is asking how many peaches Jill has. So, we need to find how many peaches Jill has. (jill)</li><li>2. The problem tells us that Steven has 19 peaches. (steven = 19)</li><li>3. The problem tells us that Steven has 13 more peaches than Jill. (jill = steven - 13)</li><li>4. The problem tells us that Jake has 18 fewer peaches than Steven. (jake = jill - 18)</li><li>5. So, the answer is <math>19 - 13 = 6</math>.</li></ol>	<p>Contains a valid solution <input type="radio"/> Yes <input type="radio"/> No</p> <p>Is complete <input type="radio"/> Yes <input type="radio"/> No</p> <p>Is correct <input type="radio"/> Yes <input type="radio"/> No</p> <p>Is relevant <input type="radio"/> Yes <input type="radio"/> No</p> <p>Is interpretable <input type="radio"/> Yes <input type="radio"/> No</p> <p><a href="#">Next</a></p> <p>You've graded 1/22 explanations so far</p>

Figure 6: The annotation interface shown to annotators to annotate explanations.

The Question:	The Answer:	The Explanation:	Does the explanation meet the criteria?
The ring toss game at the carnival made 325 dollars in the first 154 days and 114 dollars in the remaining 57 days. How much did they make per day in the remaining 57 days at the carnival?	<pre>def incorrect_solution():     return 114/(154+57)</pre>	<p>What approach does the correct solution take: The correct solution takes the total money made in the last 57 days and divides it by the 57 days. It also names the variables explicitly and expands the problem across multiple lines.</p> <p>What approach does the incorrect solution take: The incorrect solution takes the total money made in the last 57 days and divides it by the number of ducks.</p> <p>Why is the incorrect solution incorrect: You shouldn't be dividing the money made in the last 57 days by the total number of days. You should be dividing the money made in the last 57 days by the 57 days.</p>	<p>Contains a valid solution <input type="radio"/> Yes <input type="radio"/> No</p> <p>Is complete <input type="radio"/> Yes <input type="radio"/> No</p> <p>Is correct <input type="radio"/> Yes <input type="radio"/> No</p> <p>Is relevant <input type="radio"/> Yes <input type="radio"/> No</p> <p>Is interpretable <input type="radio"/> Yes <input type="radio"/> No</p> <p><a href="#">Next</a></p> <p>You've graded 1/20 explanations so far</p>

Figure 7: The annotation interface shown to annotators to annotate corrections.

# A Transfer Learning Pipeline for Educational Resource Discovery with Application in Survey Generation

Irene Li<sup>1\*</sup>, Thomas George<sup>2</sup>, Alex Fabbri<sup>1</sup>, Tammy Liao<sup>1</sup>,  
Benjamin Chen<sup>1</sup>, Rina Kawamura<sup>1</sup>, Richard Zhou<sup>1</sup>, Vanessa Yan<sup>1</sup>,  
Swapnil Hingmire<sup>3</sup> and Dragomir Radev<sup>1</sup>

<sup>1</sup>Yale University, <sup>2</sup>University of Waterloo,

<sup>3</sup>Tata Consultancy Services Limited

## Abstract

Effective human learning depends on a wide selection of educational materials that align with the learner’s current understanding of the topic. While the Internet has revolutionized human learning or education, a substantial resource accessibility barrier still exists. Namely, the excess of online information can make it challenging to navigate and discover high-quality learning materials in a given subject area. In this paper, we propose an automatic pipeline for building an educational resource discovery system for new domains. The pipeline consists of three main steps: resource searching, feature extraction, and resource classification. We first collect frequent queries from a set of seed documents, and search the web with these queries to obtain candidate resources such as lecture slides and introductory blog posts. Then, we process these resources for BERT-based features and meta-features. Next, we train a tree-based classifier to decide whether they are suitable learning materials. The pipeline achieves F1 scores of 0.94 and 0.82 when evaluated on two similar but novel domains. Finally, we demonstrate how this pipeline can benefit two applications: prerequisite chain learning and leading paragraph generation for surveys. We also release a corpus of 39,728 manually labeled web resources and 659 queries from NLP, Computer Vision (CV), and Statistics (STATS).

## 1 Introduction

People rely on the internet for various educational activities, such as watching lectures, reading textbooks, articles, and encyclopedia pages. One may wish to develop their knowledge in a familiar subject area or to learn something entirely new. Many online tools exist that enable and promote independent learning (Montalvo et al., 2018; Romero and Ventura, 2017; Fabbri et al., 2018a; Li et al., 2019). A subset of these platforms provide primary literature resources (e.g. publications), such as Google

Scholar<sup>1</sup> and Semantic Scholar<sup>2</sup>. As an alternative to these advanced materials, other educational platforms such as MOOC.org<sup>3</sup> deliver free online courses. Also, unstructured searching on the internet is a popular method to discover other useful resources, such as blog posts, GitHub projects, tutorials, lecture slides and textbooks. Rather than diving into the technical details, these secondary literature resources provide a broad overview of the given domain, which is more valuable for beginners. Still, sifting through this material can be challenging and time-consuming, even if the learner is simply looking for a general and reliable introduction into a new subject area.

Publicly accessible data repositories that focus on gathering a fixed number of educational resources exist currently, such as scientific papers (Tang et al., 2008, 2010), online platforms like AMiner (Sinha et al., 2015) and Semantic Scholar. Some archives also compile secondary literature materials. TutorialBank (Fabbri et al., 2018a) is a manually-collected corpus with over 6,300 NLP resources, as well as related fields in Artificial Intelligence (AI), Machine Learning (ML) and so on. LectureBank (Li et al., 2020) is also a manually-collected corpus and contains 1,717 lecture slides. MOOCube (Yu et al., 2020) is a large-scale data repository containing 700 MOOC (Massive Open Online Courses), 100k concepts and 8 million student behaviours with an external resource. However, in their initial synthesis, these existing corpora either heavily relied on manual efforts that restricted in certain domains, or on a large volume of existing courses sourced from a certain platform. Such solutions are not practically extensible into new or evolving domains. Moreover, according to (Fabbri et al., 2018a), some web data such as blog posts, tutorials and educational web pages are

\*Corresponding author: irene.li@aya.yale.edu

<sup>1</sup><https://scholar.google.com/>

<sup>2</sup><https://www.semanticscholar.org/>

<sup>3</sup><https://www.mooc.org/>

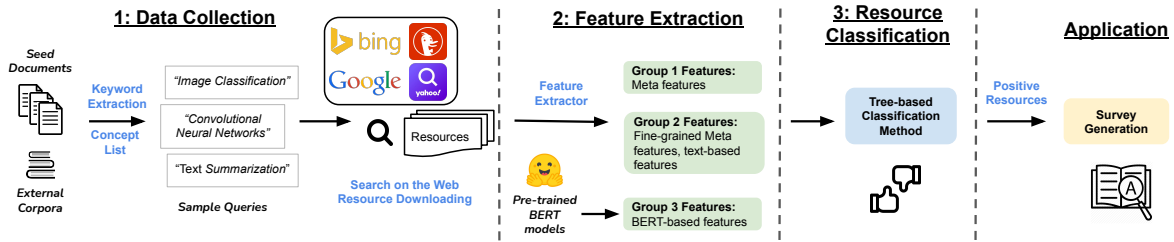


Figure 1: Pipeline Overview. The pipeline contains three steps: query generation, feature extraction, and classification & evaluation. We also show an application in this figure.

also suitable materials for learners. These rich web data are ignored by existing educational platforms such as google scholar and MOOCcube. In this paper, we wish to ease the need for human annotators by proposing a pipeline that automates resource discovery to similar unseen domains through transfer learning. Besides, such a pipeline deals with multiple resource types to take advantage of web data.

Our contributions can be summarized into three parts. First, we present a self-sustaining pipeline for educational resource discovery in close unseen subject area or domain. We apply transfer learning with a novel pre-training information retrieval (IR) model, achieving competitive performances. We show that this pipeline achieves 0.94 and 0.82 F1 scores for two arbitrary target domains on discovering high-quality resources. Second, we demonstrate an application that leverage resources discovered by our pipeline, survey generation for leading paragraph. Lastly, we release the core source code of the pipeline, as well as the training and testing datasets, comprised of 39,728 manually labelled web resources and 659 search queries. <sup>4</sup>

## 2 Educational Resource Discovery Pipeline

We propose the Educational Resource Discovery (ERD) pipeline that aims at automatically recognizing high-quality educational resources. We model this problem as a resource classification task. Given a resource  $r$ , where  $r$  can be any source type such as web page, PDF, we can obtain a list of features by feature engineering; based on these features,  $r$  is classified positive if it is a high-quality resource, otherwise negative. We illustrate the ERD pipeline in Figure 1. It consists of data collection, feature extraction and resource classification.

<sup>4</sup><https://github.com/IreneZihuiLi/Educational-Resource-Discovery>

### 2.1 Data Collection

#### 2.1.1 Queries for search

In this step, we need to conduct a list of meaningful and fine-grain search queries to start. These search queries will then be applied to online search engines for web resources. Queries can be borrowed from external corpora or extracted from existing seed documents (e.g., textbooks). We focus on three domains: NLP (natural language processing), CV (computer vision) and STATS (statistics). For NLP queries, we utilize external topic lists provided by LectureBankCD (Li et al., 2021), in which there are totally 322 NLP-based and 201 CV-based topics from crowdsourcing. For STATS, we extract a list of fine-grained terms from several seed documents, including several textbooks. These terms contain frequent keywords and phrases that are extracted by TextRank (Mihalcea and Tarau, 2004), a statistical method to keyword ranking. In total, we end up with 322, 201 and 137 queries for NLP, CV and STATS domain.

To craft our search engine queries, we leverage advanced search conditions: *filetype* and *site* (website). Specifically, we consider three file types: PDF, PPTX/PPT, and HTML. Moreover, according to the TutorialBank corpus (Fabbri et al., 2018b), resources clustered by the components of their URL possess highly correlated educational content. Thus, we prioritize restricting our queries to websites that consistently provide high-quality resources. We select the top sites from the manually-created TutorialBank corpus and incorporate them into our search queries, as exemplified in 1. We also include the “.edu” top-level domain as a special case for our search queries in order to capture general educational resources. Finally, we combine our query terms with the website and file-type constraints: e.g. “word embeddings filetype:pdf”. We also augment the original query by generating a disjunction of its variations: e.g., “stochastic gradient

towardsdatascience.com	datahacker.rs
medium.com	hackernoon.com
www.analyticsvidhya.com	skymind.ai
www.kdnuggets.com	maelfabien.github.io
machinelearningmastery.com	rubikscodex.net
paperswithcode.com	research.googleblog.com

Table 1: Top sites found in the TutorialBank corpus (Fabbri et al., 2018b).

descent” becomes “stochastic gradient descent OR SGD”. Table 2 displays several sample queries.

Once the queries are generated, we leverage three well-established online search engines: DuckDuckGo (<https://duckduckgo.com/>), Yahoo (<https://search.yahoo.com/>) and Bing (<https://www.bing.com/>) to obtain our candidate resources. The top  $N$  URLs (where  $N$  is determined from the domain, file type and site type, varying from 20 to 100 to control the total number of resources we want to collect) for a given query are cached after checking their HTTP response status and ensuring that a URL has not already been collected as part of another query. Moving forward, the documents pointed to by all of these URLs were automatically downloaded and parsed for their features. Certain features, such as the number of authors were collected using heuristics that accounted for most of the variability within the diverse dataset. The ERD Pipeline’s parsers use the pdfminer<sup>5</sup> and grobid<sup>6</sup> libraries for PDF files, Apache Tika<sup>7</sup> for PPTX/PPT and beautifulsoup<sup>8</sup> for HTML.

### 2.1.2 Annotation

After collecting all resources, the next step is to assign a binary label to each resource based on its quality. Our annotators consist of 7 graduate and senior college students with a solid background in NLP, CV, and STATS. A resource is annotated as positive if it is a high-quality one. Guidelines for a positive resource are:

- *Informative and relevant*: introducing basic knowledge about a specific topic. For example, tutorials, introductions, explanations, guides.
- *Papers and lecture slides*: papers and lecture notes about a topic in the correct domain.

<sup>5</sup><https://github.com/pdfminer/>

<sup>6</sup><https://github.com/kermitt2/grobid>

<sup>7</sup><https://tika.apache.org/>

<sup>8</sup><https://crummy.com/software/BeautifulSoup/>

### NLP Sample Queries

“morphological disambiguation ” filetype:pptx  
“word embeddings ” filetype:pdf  
“text classification tutorial ”  
“summarization nlp tutorial” site:edu

### CV Sample Queries

“computer graphics ” site:kdnuggets.com  
“texture classification ” filetype:pptx

### STATS Sample Queries

“conditional probability ” site:kdnuggets.com  
“multinomial distribution introduction ” filetype:html

Table 2: Sample queries in the three domains.

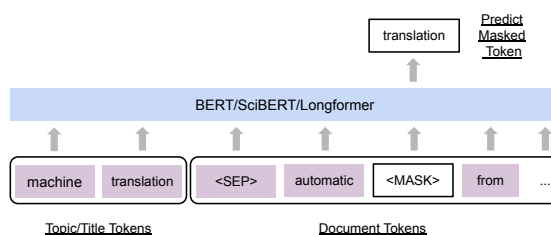


Figure 2: QD-BERT MLM pretraining.

- *Other secondary literature articles*: i.e., blog posts with informative descriptions, definitions and code blocks.

The annotation criteria for a poor resource are:

- *Not informative*: dataset/software/tool download page without introductory descriptions, such as a paper abstract page (not the paper content), a download page with links.
- *Irrelevant*: not showing correct content, broken URLs, URLs with not enough or no text (video or image only).
- *No knowledge included*: such as a course landing page, a person’s personal website page.
- *A list of resources/datasets*: containing only links to other pages.

Finally, to measure the inter-coder agreement of the labels, we randomly picked 100 resources and asked each annotator to provide labels independently. Krippendorff’s alpha (Krippendorff, 2011) on this sample evaluated to 0.8344, indicating a high degree of consistency amongst all annotators.

We detail statistics about our collected dataset in Table 2, providing the total counts by file type and domain. From the three domains, we collected 39,728 valid resources using 659 distinct queries and achieved a total positive rate of 69.05%.

	NLP	CV	STATS	Total
Query Num	322	200	137	659
PPTX	1,216	733	1,463	3,412
PDF	4,961	3,782	1,449	10,192
HTML	9,368	9,302	7,454	26,124
<b>Total</b>	<b>15,545</b>	<b>13,817</b>	<b>10,366</b>	<b>39,728</b>
Pos.Num	9,589	11,101	6,742	27,432
Pos.Rate	0.6169	0.8034	0.6501	<b>0.6905</b>

Table 3: Dataset statistics by domain and file type. *Pos.Num* is the number of positive resources. *Pos.Rate* is the fraction of resources that were labeled as positive.

## 2.2 Feature Extraction

To train a classifier to identify high-quality educational resources, we first focus on feature engineering. Specifically, we investigate the following three groups of classification features and summarize them in Table 4.

**Group 1 Features** Some of the meta-features of a document that can characterize its quality are embedded in its structure. The features encompassed by Group 1 are high-level and coarse-grained, and focus on aspects such as: the number of headings, equations, outgoing links and authors in a given resource. Heuristically, some good tutorials may tend to include more equations and paragraphs, with many details included. We list all 8 such features in Table 4, Group 1.

**Group 2 Features** These meta-features describe the fine-grained but statistical details of the document. The resource URL’s components, such as the top-level domain name and subdomain name, correlate resources from websites that deliver consistent quality. The other Group 2 features are centered around the characteristics of the free text. For instance, *NormalizedUniqueVocab* (the size of the vocabulary divided by the total number of words) can estimate the vocabulary’s complexity and *PercentTypos* (the percentage of words that are incorrectly spelled) can approximate reliability. We itemize such features in Table 4, Group 2.

**Group 3 Features** In addition to the above features, we propose 9 features based on pretrained language models. To achieve this, we first choose three models<sup>9</sup>: BERT (Devlin et al., 2019), SciBERT (Beltagy et al., 2019) and Longformer (Beltagy et al., 2020). BERT is a pretrained language model that was pretrained on Wikipedia documents. SciBERT is a BERT-based model trained on the sci-

<sup>9</sup>[https://huggingface.co/transformers/pretrained\\_models.html](https://huggingface.co/transformers/pretrained_models.html)

Feature Name	Explanation
<i>Group 1</i>	
NumAuthor	Number of authors
NumHeading	Number of headings
NumFig	Number of figures
NumEqu	Number of equations
NumPara	Number of paragraphs
NumSent	Number of sentences
NumLink	Number of outgoing links
BibLen	Bibliography length
<i>Group 2</i>	
Subdomain	Subdomain of resource URL
SecondDomain	Second-level domain of resource URL
TopDomain	Top-level domain of resource URL
NumUrlSubdirs	Number of URL subdirectories
NormalizedUniqueVocab	Number of unique words divided by total number of words
UniqueVocabMean	Mean number of occurrences of a word
UniqueVocabStdev	Stdev of number of occurrences of a word
WordLenMean	Mean number of characters per word
WordLenStdev	Stdev of number of characters per word
SentenceLenMean	Mean number of words per sentence
SentenceLenStdev	Stdev of number of words per sentence
PercentTypos	Percentage of words that were misspelled
NumGithubLinks	Number of links to GitHub
<i>Group 3</i>	
bert	BERT base model
scibert	SciBERT base model
longformer	Longformer base model
arXiv_bert	BERT pre-trained on arXiv
arXiv_scibert	SciBERT pre-trained on arXiv
arXiv_longformer	Longformer pre-trained on arXiv
TB_longformer	BERT pre-trained on TutorialBank
TB_bert	SciBERT pre-trained on TutorialBank
TB_scibert	Longformer pre-trained on TutorialBank

Table 4: Chosen features: we select 3 groups consist of meta features and deep learning-based features.

entific domain, making it suitable for our use case. Longformer is a BERT-based model that handles longer input sequences.

Moreover, we introduce a novel pre-training approach: QD-BERT MLM (Query-document BERT Masked Language Modeling). A query could be a single word, phrase or a paper title, indicating the **topic** or **main idea** of the document. We pair the query term with the corresponding document as the input and follow the Masked Language Modeling (MLM) method of BERT (randomly masking 15% tokens and letting the model predict them), as shown in Figure 2. We apply two external corpora for pre-training to ensure the data quality: TutorialBank (TB)<sup>10</sup> and arXiv<sup>11</sup>. The latest TutorialBank has 15,584 topic-document pairs; and arXiv has 259,050 title-abstract pairs (computer science papers only). We enumerate all models in Table 4, Group 3, naming *dataset\_modelname*.

We propose an information retrieval-based scoring function to combine features from deep models with Group 1 and 2 features. This scoring function

<sup>10</sup><http://aan.how/download/>

<sup>11</sup><https://www.kaggle.com/Cornell-University/arxiv>



Features	NLP→CV			NLP→STATS		
	F1	Precision	Recall	F1	Precision	Recall
Group 1	0.7238	0.5802	0.9617	0.6508	0.5405	0.8177
Group 1 + 2	0.8579	0.7772	0.9571	0.7990	0.8141	0.7845
Group 3, BERT Only*	0.7764	0.7522	0.8497	0.7923	0.7903	0.7944
Group 1 + 2 + 3	<b>0.9402</b>	0.9849	0.8994	<b>0.8225</b>	0.9965	0.7002

Table 5: Classification Results in two target domains: CV and STATS. For Group 3, BERT Only\*, we report the best model: CV (*scibert*), STATS (*TB\_scibert*).

calculates a score of each resource, showing the relevancy of the resource to all the searching queries. Relevancy is one of the most indicators that the resource is annotated as positive. The score is higher if it is more relevant to the queries. In Section 2.1.1, we apply a list of queries ( $q \in Q$ ) to download resources, we compute a cosine-similarity based ranking score  $score_r$  for resource  $r$ :

$$score_r = \sum_{q \in Q} cosine(V_q, V_r)$$

where  $V_q$  and  $V_r$  are BERT-based model embeddings for the query term and resource respectively. We compute scores on each pre-trained BERT models of each resource.

### 2.3 Resource Classification

Since there are various feature types, we conduct preprocessing before applying the classifiers. Numerical values are binned into groups, and categorical features are converted into integer codes. We evaluate four traditional classifiers: Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM) and Logistic Regression (LR). We find that RF performs the best and has a slight edge over DT, but SVM and LR significantly lag behind. Thus, we report the Random Forest’s performance, summarized in Table 5. Specifically, we include precision, recall and F1 scores on different feature groups: Group 1, Group 1+2, and Group 1+2+3. The last setting achieves the best performance. Additionally, since it is also possible to solely apply BERT models (Group 3) for the classification task, we include a special setting: Group 3, BERT only. While BERT’s results in isolation are good, Group 1+2+3 still remains the winner.

In general, performance on the CV domain is better than on STATS. This is expected given that the corpus distance between NLP and CV is smaller than the one between NLP and STATS. We give detailed data analysis in the next section.

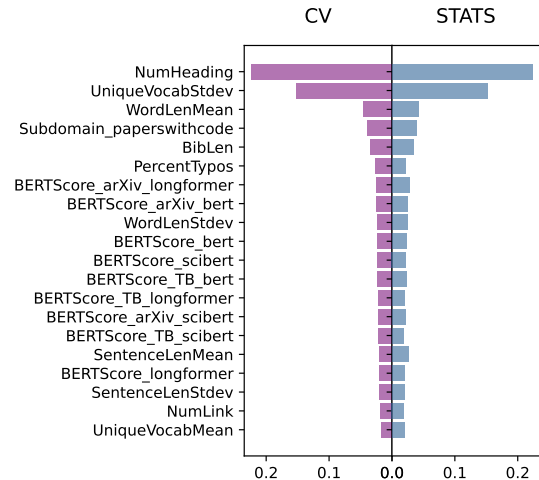


Figure 3: Top 20 features on two target domains.

## 3 Data Analysis

To better understand the collected data and our classifier’s performance, we conduct a study on the features and corpus differences between the three experimental domains.

**Feature Importance Score** We take the best-performed model of NLP→CV domain (Group 1+2+3), and take the Gini Index calculated by Decision Trees as the feature importance score. Overall, we extract 8746 features in CV and 8525 features of STATS after binning numerical values and encoding categorical features. In Figure 3, we list the top 20 features of CV and STATS. Some Group 1+2 features rank in the top 5, since they are main indicators that the resource is informative (i.e., more heading numbers, longer contents). Additionally, Group 3 features (starting with *BERTScore*) also play an important role. In fact, all 9 BERT-based feature scores rank top 20, suggesting that our scoring function that adds these BERT-based semantic features into the pipeline is very helpful when doing classification for resource discovery.

**Corpus Differences** Our pipeline performs better on CV topics, which can be attributed to cor-

Domain	Top 10 Sites
NLP	<a href="http://www.cs.cmu.edu">www.cs.cmu.edu</a> , <a href="http://web.stanford.edu">web.stanford.edu</a> , <a href="http://www.cs.toronto.edu">www.cs.toronto.edu</a> , <a href="http://www.paperswithcode.com">www.paperswithcode.com</a> , <a href="http://maelfabien.github.io">maelfabien.github.io</a> , <a href="http://www.academia.edu">www.academia.edu</a> , <a href="http://courses.cs.washington.edu">courses.cs.washington.edu</a> , <a href="http://nlp.stanford.edu">nlp.stanford.edu</a> , <a href="http://ocw.mit.edu">ocw.mit.edu</a> , <a href="http://www.cs.cornell.edu">www.cs.cornell.edu</a>
CV	<a href="http://www.kdnuggets.com">www.kdnuggets.com</a> , <a href="http://maelfabien.github.io">maelfabien.github.io</a> , <a href="http://www.paperswithcode.com">www.paperswithcode.com</a> , <a href="http://www.academia.edu">www.academia.edu</a> , <a href="http://www.cs.toronto.edu">www.cs.toronto.edu</a> , <a href="http://www.cs.cmu.edu">www.cs.cmu.edu</a> , <a href="http://web.stanford.edu">web.stanford.edu</a> , <a href="http://courses.cs.washington.edu">courses.cs.washington.edu</a> , <a href="http://cseweb.ucsd.edu">cseweb.ucsd.edu</a> , <a href="http://www.cs.cornell.edu">www.cs.cornell.edu</a>
STATS	<a href="http://www.kdnuggets.com">www.kdnuggets.com</a> , <a href="http://maelfabien.github.io">maelfabien.github.io</a> , <a href="http://www.paperswithcode.com">www.paperswithcode.com</a> , <a href="http://web.stanford.edu">web.stanford.edu</a> , <a href="http://ocw.mit.edu">ocw.mit.edu</a> , <a href="http://online.stat.psu.edu">online.stat.psu.edu</a> , <a href="http://www.hackernoon.com">www.hackernoon.com</a> , <a href="http://www.sjsu.edu">www.sjsu.edu</a> , <a href="http://research.googleblog.com">research.googleblog.com</a> , <a href="http://www.cpp.edu">www.cpp.edu</a>

Table 6: Comparison of the top 10 sites. **Gray** means overlapped in both CV and STATS domain; **Purple** means overlapping between NLP and CV; **Blue** means overlapping between NLP and STATS.

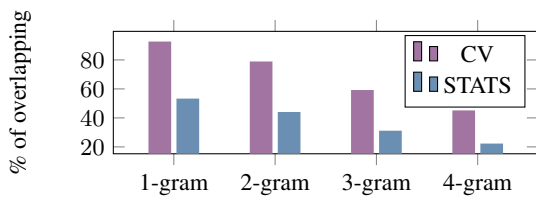


Figure 4: Percentage of overlapping n-grams.

pus differences relative to NLP. In Figure 4, we plot the percentage of overlapping n-grams of the {NLP, CV} and {NLP, STATS} domain pairs. This shows that NLP and CV have a larger overlap than {NLP, STATS} with respect to all of the n-grams ( $n \in \{1, 2, 3, 4\}$ ). From this, we uphold that the classifiers trained on semantic features based on BERT models are valuable for bridging more distant domains with transfer learning.

To further contrast our findings, we enumerate the top 10 URLs in Table 6. Although the websites are ranked in different orders, there are still common URLs across the domains (highlighted in the table). Once again, CV shares a larger overlap with NLP in comparison to STATS. Along with the feature importance score, this cross-domain consistency further illustrates that the URL meta-features will benefit our model’s out-of-domain classification. We show more feature statistics in the Appendix.

**Comparison With Similar Datasets** We compare a number of existing NLP educational datasets in Table 7, emphasizing the resource type, human effort for annotations, and corpus scale. Note that in this table, we only concentrate on human annotation efforts for free-text resources. This is because these free-text resources are the primary goal of the ERD Pipeline, as opposed to other tasks (e.g. learning concept relations, concept mining). We can see that MOOCcube (Yu et al., 2020) has a massive

quantities of a single resource type (papers). They obtained the metadata from a third-party platform, AMiner, without a full round of human annotations. TutorialBank (Fabbri et al., 2018b) has a larger number of resources than LectureBank (Li et al., 2020), and it consists of diverse resource types. Our pipeline is very similar to TutorialBank in terms of resource type, but ours extends to more resources and subject areas, enabling us to research transfer learning across domains.

#### 4 Application: Survey Generation for Lead Paragraphs

In this section, we demonstrate an interesting application that applies the resources discovered using our ERD Pipeline, Leading Paragraph Generation for Surveys.

Novel concepts are being introduced and evolving at a rate that creates high-quality surveys for web resources, such as Wikipedia pages, challenging. Moreover, such existing surveys like Wikipedia still needs human efforts on collecting relevant resources and writing accurate content on a given topic. Researchers have been investigating automatic ways to generate surveys using machine learning and deep learning methods. Survey generation is a way to generate concise introductory content for a query topic (Zhao et al., 2021). While most of the existing work focuses on utilizing Wikipedia to achieve this (Liu et al., 2018), little has been done for the web content. Since our ERD pipeline provides sufficient web data, we propose a two-stage approach for generating the lead paragraph that applies these web data selected from the ERD pipeline.

Name	Resource Type (with texts)	Domain Number	Annotation	Size
TutorialBank	Lecture sides, papers, blog posts	NLP only	Manually	6,300
LectureBank	Lecture sides only	NLP only	Manually	1,717
MOOCcube	Papers only	Multiple	Scrape from third-party	679,790
ERD (ours)	Lecture sides, papers, blog posts	Multiple	Manually	39,728

Table 7: Comparison with similar datasets.

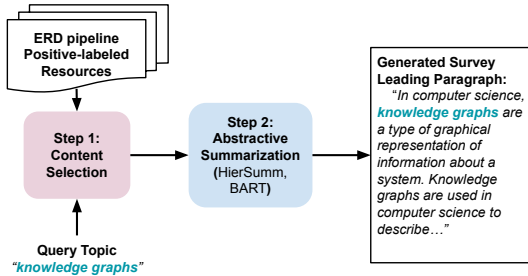


Figure 5: Two-stage Survey Generation Method.

#### 4.1 Two stage method

We illustrate the two stage method in Figure 5. Given a query topic and high-quality web resources selected by ERD pipeline, we wish to generate the leading introductory paragraph for the query topic. This approach consists of content selection (step 1) and abstractive summarization (step 2). Content selection is the process of selecting the most relevant materials (including documents or sentences) according to the given query. Abstractive summarization generates the accurate lead paragraph from the selected materials.

**Content Selection** ERD pipeline is supposed to identify massive resources with broad coverage of the topics, so the first step is to select related content with the query topic.

While there is no suitable pretrained data for this task, and we do not collect survey data for training, we utilize the WikiSum dataset (Liu et al., 2018).

Methods	L=5	L=10	L=20	L=40
LSTM-Rank	39.38	46.74	53.84	60.42
Semantic Search	34.87	48.60	61.87	74.54
RoBERTa-Rank	<b>64.12</b>	<b>72.49</b>	<b>79.17</b>	<b>84.28</b>

(a) ROUGE-L (Lin, 2004) Recall scores for WikiSum content selection, varying the number of paragraphs returned.

Methods	R-1	R-2	R-L
HierSumm (Liu and Lapata, 2019)	41.53	26.52	35.76
BART (Lewis et al., 2019)	<b>46.61</b>	<b>26.82</b>	<b>43.25</b>

(b) ROUGE scores for intro generation.

Table 8: Two-stage method evaluation using WikiSum.

WikiSum contains 1.5 million Wikipedia pages, their references and their associated Google Search results. WikiSum includes many well-established topics and comprehensive reference documents, making it suitable for survey generation. We first evaluate content selection models using WikiSum. We experiment with three approaches in this step. Liu and Lapata (2019) undertake query-based content selection as a regression problem of predicting the ROUGE-2 recall of a given paragraph-topic pair (LSTM-Rank). Reimers and Gurevych (2019) fine-tune BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) to produce fixed-length vectors which can be compared using cosine similarity. We embed the topic of each Wikipedia page and candidate paragraph using this method, and select the paragraphs with the closest vectors to the title (Semantic Search). Additionally, we train RoBERTa in a similar manner as (Liu and Lapata, 2019). Then, we compare the query topic and paragraphs as sentence pairs and use the resultant relevance scores to for the paragraph ranking (RoBERTa-Rank). As shown in Table 8a, RoBERTa-Rank is the highest-scoring content selector, so we employ it for the abstractive summarization’s input.

**Abstractive Summarization** This step is to generate summarization from the content selected previously. As a sequence-to-sequence task, there are many existing pretrained models to use. We experiment with BART (Lewis et al., 2019), a pre-trained model for text generation, as well as HierSumm, a hierarchical model from Liu and Lapata (2019). We show the summarization results on the WikiSum data in Table 8b, and observe that BART achieves the higher performance.

Evaluation	NLP	CV	STATS
Avg. Readability	3.45	2.90	2.75
Avg. Relevance	2.80	1.85	1.50
Avg. Non-redundancy	2.45	2.15	1.60

Table 9: Human Evaluation

---

**NLP: knowledge graphs**

In computer science, knowledge graphs are a type of graphical representation of information about a system. Knowledge graphs are used in computer science to describe the structure of a system, such as a graph of knowledge about a particular system, ... or a list of items that can be used to describe a system's structure. For example, a knowledge graph may contain information about an object's shape, its size, its shape relative to other objects, its orientation, and its orientation relative to a point in space, or its position relative to another point in time, or any combination of these things. Knowledge Graphs can also be used as a way of describing the structure and structure of systems, such that a system can be described by a set of knowledge graphs that describe its structure and its properties.

---

**CV: image retrieval**

In computer science, a Convolutional neural network is a type of neural network used in image analysis and image synthesis. It is a computer program that uses a neural network to learn from a large number of images, and can be used to predict the appearance of a given image in real-world situations, such as the shape of a surface, or the color of an object in a 3D environment such as an image of a person's face. It can also be used for image synthesis and image analysis, as well as image processing and image processing for computer vision and image recognition. It has been used in a number of applications such as computer vision, image processing, image recognition, and computer vision for image recognition and image rendering.

---

**STATS: normal distributions**

In physics, normal distributions are a family of mathematical models that describe the distribution of normal distributions. They are used in the fields of statistics, physics, and computer science, and have been used in a wide variety of applications, including computer vision, image analysis, computer graphics, computer vision, computer graphics processing units (CPGs), as well as in the field of computer vision. They have also been used to study the visual appearance of real-world surfaces, such as the Phong Reflection Model, the Oren-Nayar model, the Koenderink et al. representation, and the Shading of the Sphere Model, and in the study of light sources and light sources in OpenGL and OpenGL-based graphics renderers.

---

Table 10: Examples of generated leading paragraphs.

## 4.2 Human Evaluation and Case Studies

So far we have shown that applying RoBERTa-Rank and BART as a two-step method gives promising results evaluated on the WikiSum dataset. We connect our pipeline with this method to generate the leading paragraph. We choose 10 queries randomly as survey topics in each domain, for example, "sentiment analysis" in NLP. A full query topic list is in the Appendix. Since we do not have ground truth, we conduct human evaluation and case studies.

We evaluate the model outputs on a 1-5 Likert scale based on the following qualities:

- *Readability*: attains a maximum score of 5 if the output is readable with a high degree of fluency and coherency.
- *Relevancy*: attains a maximum score of 5 if the output is perfectly relevant to the current topic with no hallucinations.
- *Non-redundancy*: attains a maximum score of 5 if the output has no repeating phrases/concepts.

We report average scores among 2 human judges of all topics by domain, shown in Table 9. The scores of NLP are the highest for all qualities, and STATS performed most poorly. This discrepancy may be caused by data collection bias, as more NLP resources were included.

We randomly pick one case study from each domain in Table 10. The model is able to generate leading paragraphs in a similar Wikipedia article style by giving a definition of a certain concept, following by descriptions of possible applications. Overall, while these surveys contains some facts, the quality can still be improved. For instance, the STATS paragraph exhibits some redundancy (e.g., "computer graphics", "computer vision"). As an initial experiment, we have demonstrated the opportunities of extending our ERD Pipeline to produce survey paragraphs. In the future, we aim to enhance the generated lead paragraphs and extend the model for generating complete surveys.

## 5 Conclusion

In this paper, we proposed a pipeline for automatic knowledge discovery in novel domains. We applied transfer learning with a novel MLM pre-training method and achieved competitive classification performances. Moreover, we demonstrated two applications that take advantage of resource discovered by our pipeline. Finally, we released our source code and the datasets that we collected, including the 39,728 manually labelled web resources and 659 search queries. We plan to make this pipeline an online live educational tool for the public.

## References

- Iz Beltagy, Arman Cohan, and Kyle Lo. 2019. [Scibert: Pretrained contextualized embeddings for scientific text](#). *CoRR*, abs/1903.10676.
- Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. [Longformer: The long-document transformer](#). *CoRR*, abs/2004.05150.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL*.
- Alexander Fabbri, Irene Li, Prawat Trairatvorakul, Yijiao He, Weitai Ting, Robert Tung, Caitlin Westfield, and Dragomir Radev. 2018a. [TutorialBank: A manually-collected corpus for prerequisite chains, survey extraction and resource recommendation](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 611–620, Melbourne, Australia. Association for Computational Linguistics.
- Alexander R Fabbri, Irene Li, Prawat Trairatvorakul, Yijiao He, Wei Tai Ting, Robert Tung, Caitlin Westfield, and Dragomir R Radev. 2018b. Tutorialbank: A manually-collected corpus for prerequisite chains, survey extraction and resource recommendation. In *Proceedings of ACL*. Association for Computational Linguistics.
- Klaus Krippendorff. 2011. Computing krippendorff’s alpha-reliability.
- 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*.
- Irene Li, Alexander Fabbri, Swapnil Hingmire, and Dragomir Radev. 2020. [R-VGAE: Relational-variational graph autoencoder for unsupervised prerequisite chain learning](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1147–1157, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Irene Li, Alexander R. Fabbri, Robert R. Tung, and Dragomir R. Radev. 2019. [What should I learn first: Introducing lecturebank for NLP education and prerequisite chain learning](#). In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6674–6681. AAAI Press.
- Irene Li, Vanessa Yan, Tianxiao Li, Rihao Qu, and Dragomir Radev. 2021. Unsupervised cross-domain prerequisite chain learning using variational graph autoencoders. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- 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.
- Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*.
- Yang Liu and Mirella Lapata. 2019. Hierarchical Transformers for Multi-document Summarization. *ACL*.
- 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*.
- Rada Mihalcea and Paul Tarau. 2004. [TextRank: Bringing order into text](#). In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.
- Soto Montalvo, Jesus Palomo, and Carmen de la Orden. 2018. Building an educational platform using nlp: A case study in teaching finance. *J. UCS*, 24(10):1403–1423.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Cristóbal Romero and Sebastián Ventura. 2017. Educational data science in massive open online courses. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(1):e1187.
- Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darin Eide, Bo-june Paul Hsu, and Kuansan Wang. 2015. An overview of microsoft academic service (mas) and applications. In *Proceedings of the 24th international conference on world wide web*, pages 243–246. ACM.
- Jie Tang, Limin Yao, Duo Zhang, and Jing Zhang. 2010. A combination approach to web user profiling. *ACM TKDD*, 5(1):1–44.
- Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: Extraction and mining of academic social networks. In *KDD’08*, pages 990–998.

Jifan Yu, Gan Luo, Tong Xiao, Qingyang Zhong, Yuquan Wang, Wenzheng Feng, Junyi Luo, Chenyu Wang, Lei Hou, Juanzi Li, Zhiyuan Liu, and Jie Tang. 2020. [MOCCube: A large-scale data repository for NLP applications in MOOCs](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3135–3142, Online. Association for Computational Linguistics.

Mingjun Zhao, Shengli Yan, Bang Liu, Xinwang Zhong, Qian Hao, Haolan Chen, Di Niu, Bowei Long, and Weidong Guo. 2021. [QBSUM: A large-scale query-based document summarization dataset from real-world applications](#). *Comput. Speech Lang.*, 66:101166.

## A Chosen topics for Human Evaluation in Survey Generation

Table 11 shows the randomly selected topics for survey generation, 10 from each domain.

<b>NLP</b>
adam optimizer
lstm model
dropout neural networks
recursive neural network
convolutional neural network
automatic summarization
sentiment analysis
attention mechanism deep learning
Pre-trained Language Models NLP
knowledge graphs
<b>CV</b>
transfer learning
convolutional neural network
image retrieval
image classification
feature learning
seq2seq
transformers
visual question answering
conditional probability
k means
<b>STATS</b>
linear regression
hypothesis testing
conditional probability
multinomial distribution
probability density
density estimation
normal distributions
bernoulli distribution
standard deviation
z-score

Table 11: Topics selected for human evaluation.

## B More Sample Queries

We list more sample queries in Table 12, such queries are applied in the Data Collection step of the proposed pipeline.

### NLP Sample Queries

“markov decision processes” site:.edu filetype:.pdf  
 “sentiment analysis” site:.edu filetype:.pptx  
 “unlexicalized parsing” site:kdnuggets.com filetype:.html  
 “semantic parsing” site:.edu filetype:.pdf  
 “information retrieval” site:.edu filetype:.pptx  
 “monte carlo methods” site:rubikscore.net filetype:.html  
 “natural language processing intro” site:.edu filetype:.pdf  
 “sequence to sequence” site:.edu filetype:.pptx  
 “naive bayes” site:paperswithcode.com filetype:.html  
 “latent dirichlet allocation” site:.edu filetype:.pdf

### CV Sample Queries

“epipolar geometry” site:.edu filetype:.pptx  
 “particle filters” site:hackernoon.com filetype:.html  
 “image registration” site:.edu filetype:.pdf  
 “reflectance model” site:.edu filetype:.pptx  
 “shading analysis” site:skymind.ai filetype:.html  
 “imaging geometry and physics” site:.edu filetype:.pdf  
 “texture classification” site:.edu filetype:.pptx  
 “gibbs sampling” site:kdnuggets.com filetype:.html  
 “image thresholding” site:.edu filetype:.pdf  
 “region adjacency graphs” site:.edu filetype:.pptx

### STATS Sample Queries

“linear regression” site:rubikscore.net filetype:.html  
 “hypothesis testing” site:.edu filetype:.pdf  
 “heteroscedasticity” site:.edu filetype:.pptx  
 “random event” site:paperswithcode.com filetype:.html  
 “maximum likelihood” site:.edu filetype:.pdf  
 “granger causality” site:.edu filetype:.pptx  
 “probability” site:hackernoon.com filetype:.html  
 “random sampling” site:.edu filetype:.pdf  
 “correlation coefficient” site:.edu filetype:.pptx  
 “chi-squared statistic” site:skymind.ai filetype:.html

Table 12: More sample queries used in the three selected domains, varying site and file type.

## C BERT models for Group 3 features

The three main deep features were extracted using the following pre-trained models:

### BERT-base

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

### SciBERT

[https://huggingface.co/allenai/scibert\\_scivocab\\_uncased](https://huggingface.co/allenai/scibert_scivocab_uncased).

### Longformer

<https://huggingface.co/allenai/longformer-base-4096>.

## D More Data Statistics

In Table 13, we show token-level and sentence-level statistics of our collected data.

	NLP	CV	STATS
<i>Token Number/per sentence</i>			
Mean	18.28	26.37	23.28
Median	12	19	18
Max	2,302	458,363	20,066
<i>Sentence Number</i>			
Mean	161.60	122.49	107.32
Median	55	46	52
Max	5,929	21,301	52,793

Table 13: Free text statistics by domain.

## E Meta-Feature Distributions

In the following pages, we show the histograms of the 18 quantitative meta-features collected for each data point. Recall from Table 4 that these features were segregated into two groups. Group 1 features are higher-level and generally pertain to the document layout. Group 2 features focus on more specific aspects of the resource’s URL and free text.

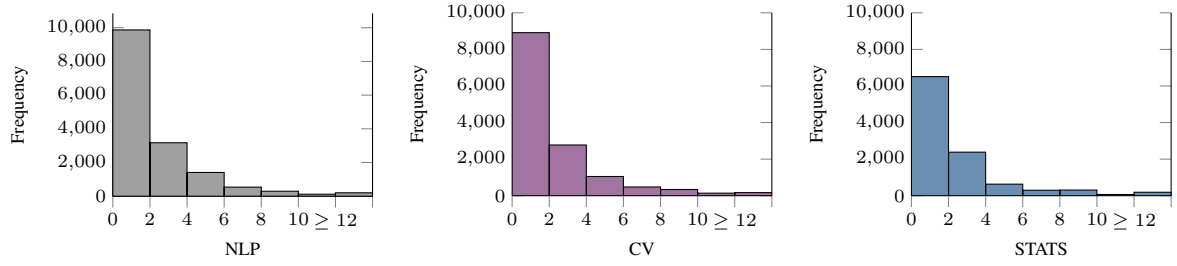


Figure 6: *NumAuthor* Distribution

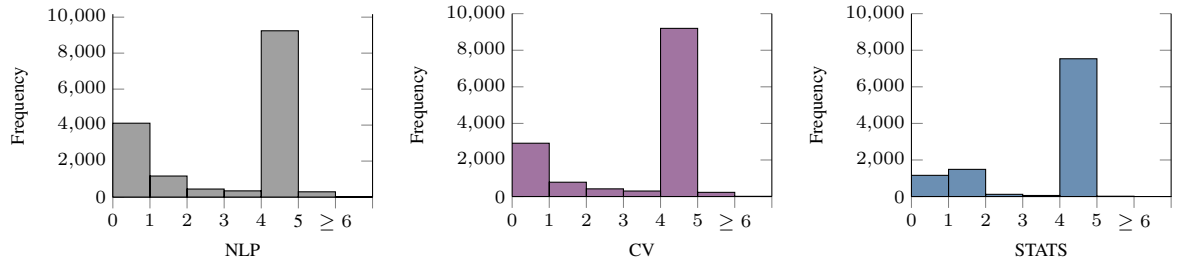


Figure 7: *NumHeading* Distribution

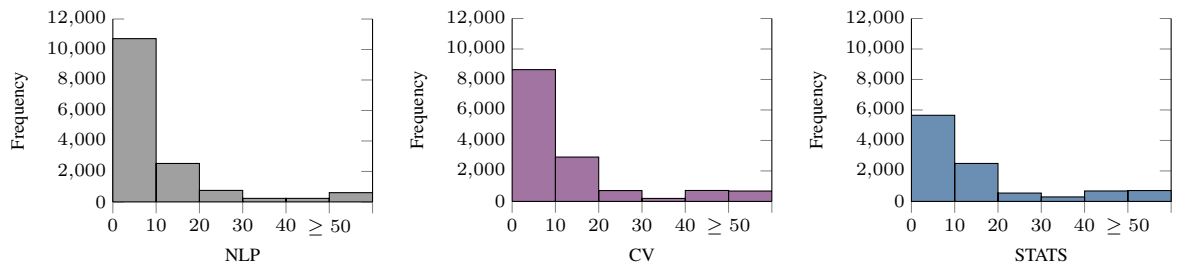


Figure 8: *NumFig* Distribution

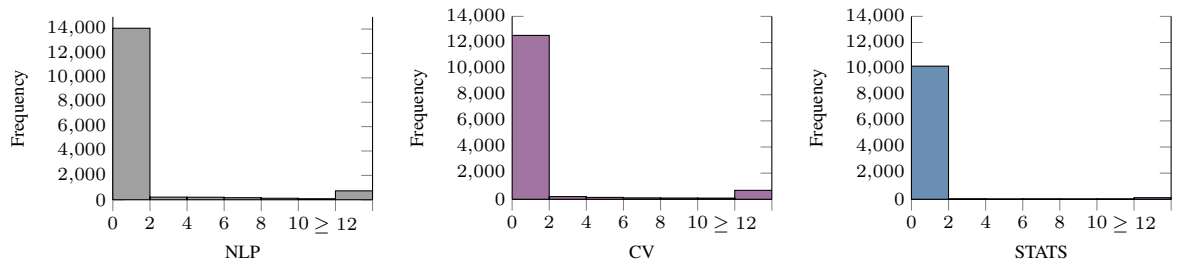


Figure 9: *NumEqu* Distribution

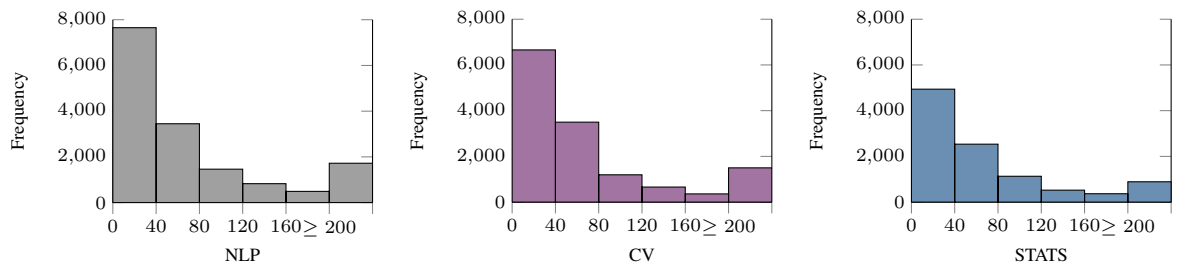


Figure 10: *NumPara* Distribution



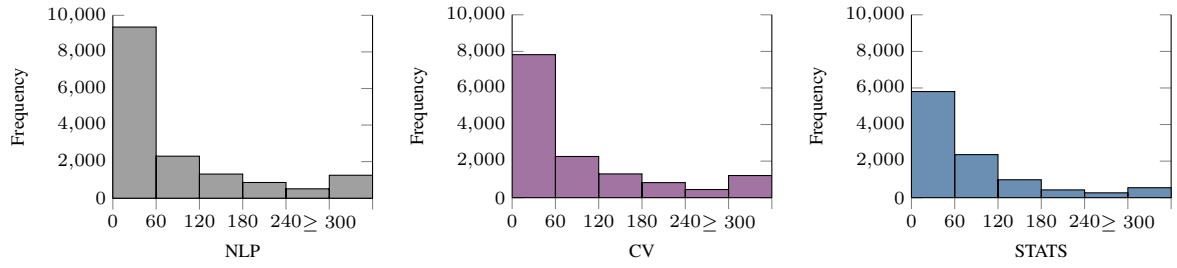


Figure 11: *NumSent* Distribution

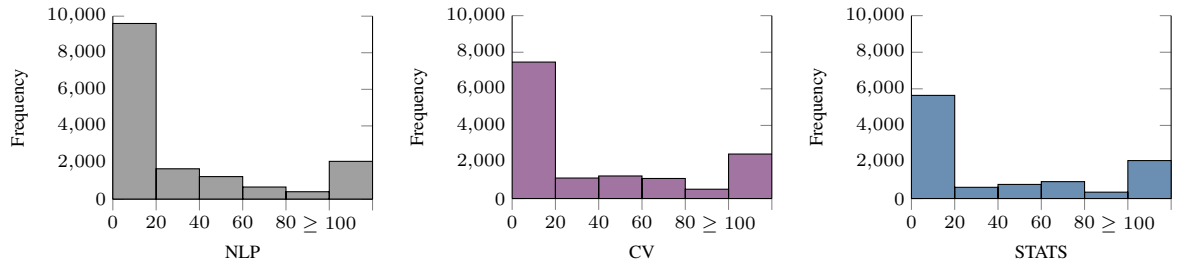


Figure 12: *NumLink* Distribution

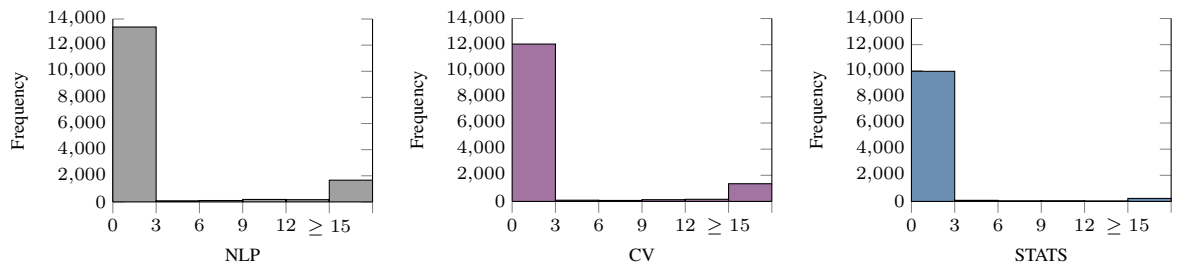


Figure 13: *BibLen* Distribution

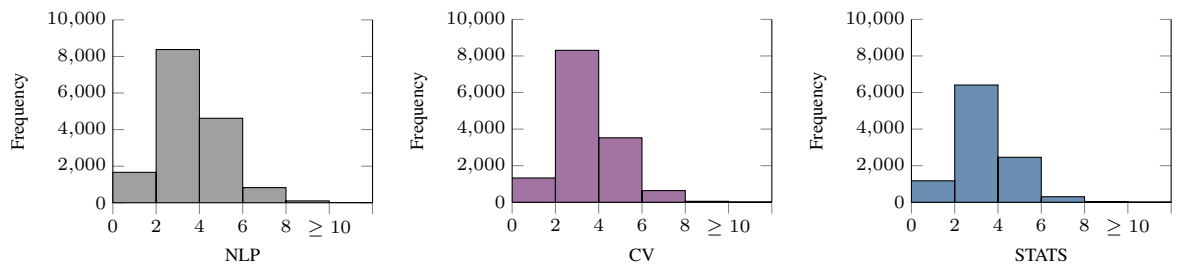


Figure 14: *NumUrlSubdir* Distribution

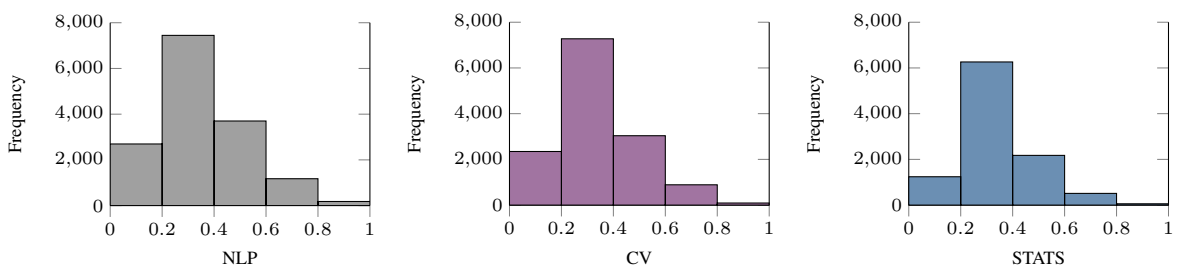


Figure 15: *NormalizedUniqueVocab* Distribution

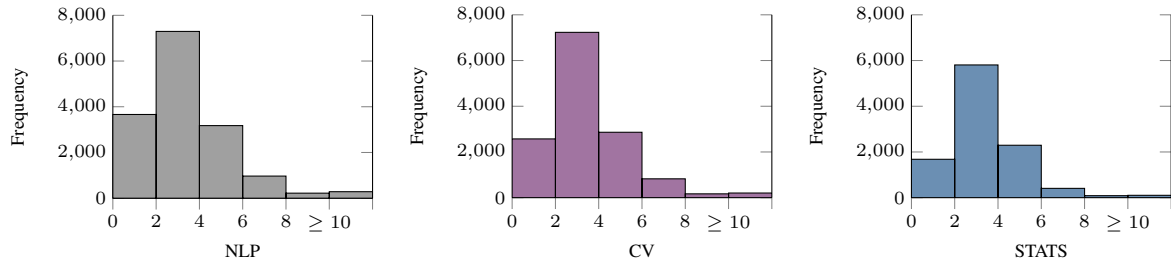


Figure 16: *UniqueVocabMean* Distribution

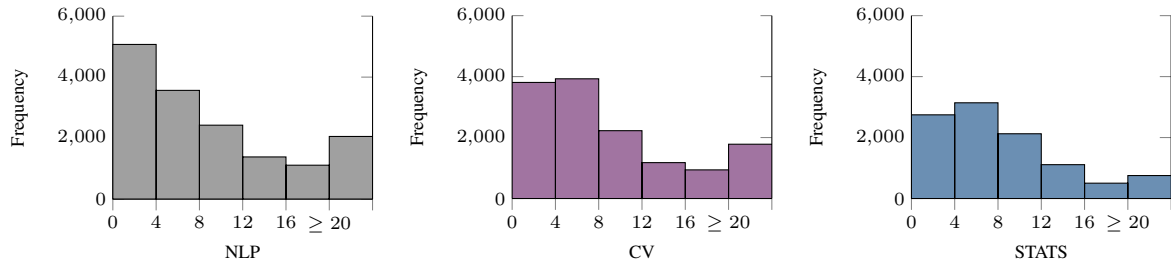


Figure 17: *UniqueVocabStdev* Distribution

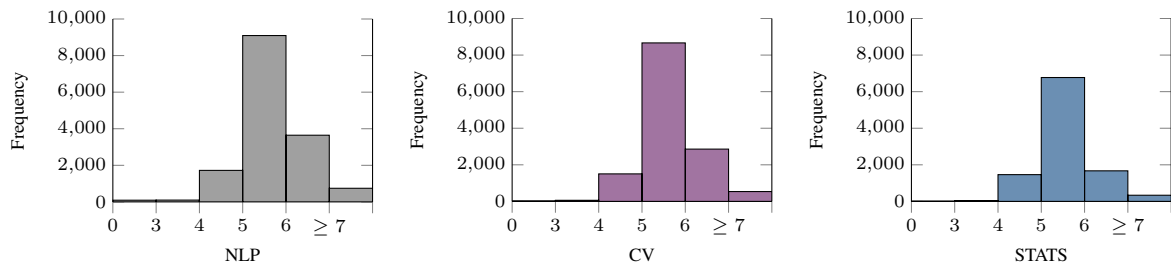


Figure 18: *WordLenMean* Distribution

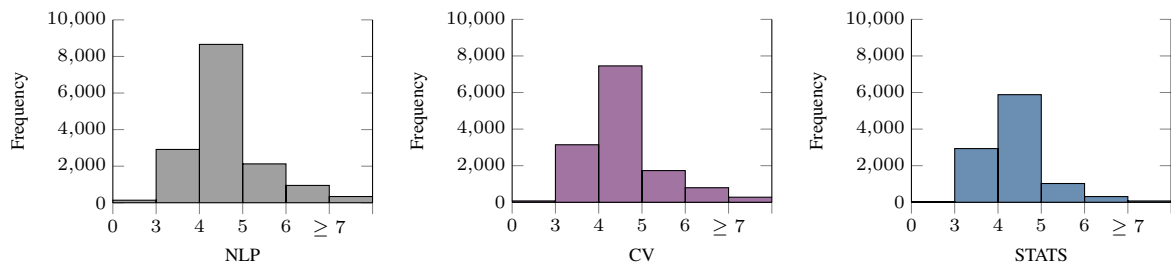


Figure 19: *WordLenStdev* Distribution

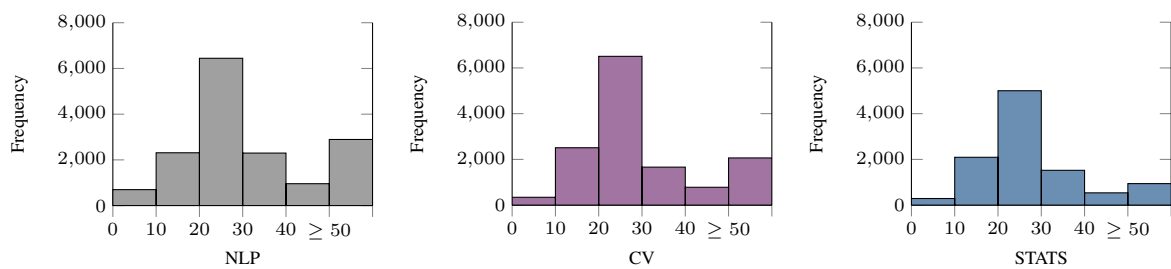


Figure 20: *SentLenMean* Distribution

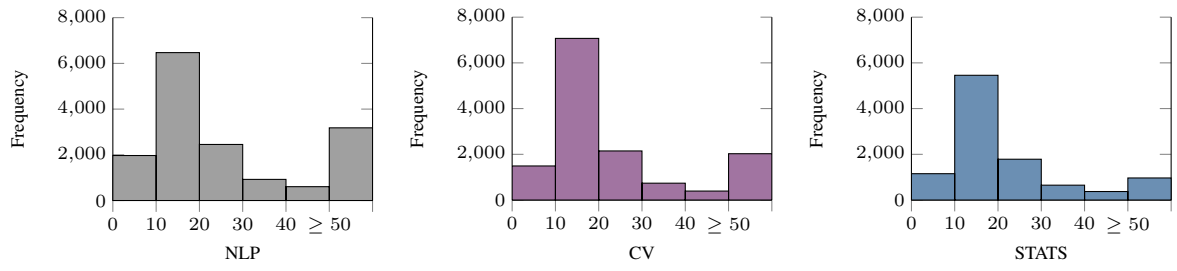


Figure 21: *SentLenStdev* Distribution

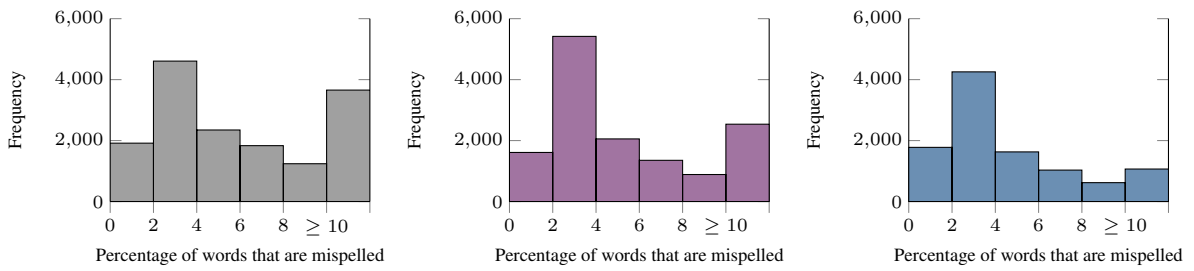


Figure 22: *PercentTypo* Distribution

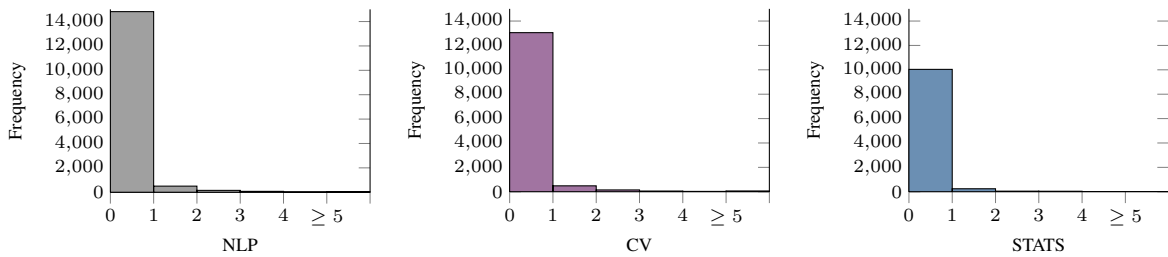


Figure 23: *NumGithubLink* Distribution

# Using Learning Analytics for Adaptive Exercise Generation

**Tanja Heck**

Universität Tübingen / Germany  
tanja.heck@  
uni-tuebingen.de

**Detmar Meurers**

Universität Tübingen / Germany  
detmar.meurers@  
uni-tuebingen.de

## Abstract

Single Choice exercises constitute a central exercise type for language learning in a learner’s progression from mere implicit exposure through input enhancement to productive language use in open exercises. Distractors that support learning in the individual zone of proximal development should not be derived from static analyses of learner corpora, but rely on dynamic learning analytics based on half-open exercises. We demonstrate how a system’s error diagnosis module can be re-used for automatic and dynamic generation and adaptation of distractors, as well as to inform exercise generation in terms of relevant learning goals and reasonable chunking in Jumbled Sentences exercises.

## 1 Introduction

Supporting language learners to progress in their zone of proximal development requires exercises of different complexities (Shabani et al., 2010). While input enhancement for implicit exposure to linguistic constructions can foster receptive skills at the lower end of the complexity range (Meurers et al., 2010), open exercises that elicit production of linguistic constructions and the entire sentence context constitute the other extreme (Becker and Roos, 2016). In order to advance from one to the other, learners need to acquire the constructions relevant for language production in a controlled way. To this purpose, half-open exercises require learners to produce only the target form whereas closed exercise types provide a range of answer alternatives to choose from (Spada and Tomita, 2010). The closed-type Single Choice (SC) exercises require special attention as they expose learners to incorrect linguistic material in the form of distractor options. While distractors should cover developmental misconceptions in order to be sufficiently challenging and thus relevant to learning, they should not expose learners to any misconceptions they would not have come up with on their own (Yamada, 2019).

Given these considerations, it is not surprising that distractor generation is seen as the most challenging aspect of generating SC exercises (Mitkov et al., 2006). In order to determine pedagogically valid and plausible distractors, human judgement is often deemed best (Susanti et al., 2018), yet even manually created distractors do often not meet these requirements (Haladyna and Downing, 1993; Patil et al., 2016). In order to automate distractor generation and at the same time increase plausibility and validity, data-driven approaches base distractors on common misconceptions of learners (Lee et al., 2016). This in addition allows a more learner-centered adaptation of distractors by dynamically selecting those distractors for each learner from a pool of options that target their individual misconceptions.

However, abstracting learner errors into patterns that facilitate generating distractors for arbitrary target answers is not a trivial task. On the other hand, many Intelligent Language Tutoring Systems (ILTS) incorporate error diagnosis mechanisms. Approaches anticipating the most common correct and incorrect learner answers, henceforth referred to as answer hypotheses, and matching them to error diagnoses (Meurers, 2012), are particularly interesting for distractor generation. An example that successfully pursues this approach constitutes the ILTS *FeedBook* (Rudzewitz et al., 2018). The process shows strong similarities to distractor generation: The most frequent learner errors constitute the most plausible distractors whereas alternative, correct answers represent unreliable distractors that need to be avoided. Systems generating answer hypotheses for error diagnoses therefore inherently have the means to automatically generate distractors. This is especially valuable if SC exercises are used for remedial practice as it opens the possibility to directly associate SC exercises with learner errors and select exercises that best target the learner’s misconceptions. The parallels of error

analysis based on answer hypotheses and distractor generation are striking, yet these two subfields of Natural Language Processing (NLP) have never been approached in tandem.

Although previous approaches to exercise generation have used learner errors solely for distractor generation, they can similarly inform chunking of Jumbled Sentences for word order practice, and determination of required exercise material. Grammatical constructions that are not challenging for learners do not need excessive practice. On the other hand, constructions where learners make many errors should be practiced in a variety of exercises focusing on remedying these misconceptions.

In order to fill the gap, we show the feasibility of using a system's error diagnosis mechanism for distractor generation, as well as for sentence chunking and learning goal definition, at the example of real learner data collected in the Interact4School (I4S) study (Parrisius et al., 2022a,b).

The rest of the paper is structured as follows: Section 2 presents related work on distractor generation. After outlining the research questions and the approach to answer them in section 3, section 4 introduces the data on which the approach was piloted. Section 5 describes the pilot analyses and presents their results before section 6 concludes with a summary.

## 2 Related work

Distractor generation usually consists of candidate generation and candidate filtering and/or ranking, although they are sometimes executed in a single step. Many approaches combine a number of different filtering and re-ranking approaches.

For question answering and vocabulary-focused gap exercises, approaches differ in the source from which the pool of distractor candidates is compiled, as well as in the filtering and ranking strategies. The candidates are either extracted from unstructured data such as text corpora (Quan et al., 2018; Gates, 2011), from structured data such as databases (Karamanis et al., 2006; Smith et al., 2009) or word lists (Coniam, 1997; Shei, 2001), or else generated based on machine learning (Liang et al., 2017; Sakaguchi et al., 2013) or on transformation rules (Žitko et al., 2009). The candidate pool then comprises either a subset (Sumita et al., 2005; Stasaski and Hearst, 2017) or all entries (Smith et al., 2010; Pérez and Cuadros, 2017) of the resource, or transformations thereof (Mar-

itxalar et al., 2011; Quan et al., 2018) or of the target answer (Zesch and Melamud, 2014). Filtering and ranking depend on the intended distractor type such as ungrammatical, nonsensical and plausible distractors (Mostow and Jang, 2012), which determines for example the usefulness of grammaticality checks (Pino et al., 2008; Moser et al., 2012). For plausible distractors, the desired similarity of the distractors with the target answer constitutes an additional factor. This is on the one hand influenced by the task setup as for example synonyms may be context-inappropriate and therefore useful distractors for contextualized exercises (Knoop and Wilske, 2013), yet would constitute unreliable distractors if they can correctly replace the target answer (Hill and Simha, 2016). In addition, since exercise difficulty increases with distractor plausibility, target similarity can be adjusted according to the learner's proficiency (Alsubait et al., 2015; Chen et al., 2015; Correia et al., 2012). Similarity can target the surface form (Jiang and Lee, 2017), linguistic complexity (Lee and Seneff, 2007; Susanti et al., 2018), phonetics (Mitkov et al., 2009), morphology (Goto et al., 2010), syntax (Guo et al., 2016), or semantics (Susanti et al., 2015) and be based on NLP tools including part-of-speech taggers (Liu et al., 2005), latent semantic analysis (Aldabe and Maritxalar, 2014) and word embedding models (Kumar et al., 2015; Yeung et al., 2019), on external resources such as ontologies (Papasalouros et al., 2008), WordNet (Mitkov et al., 2006; Brown et al., 2005) or FrameNet (Pilán and Volodina, 2014), or else on statistical methods including classification (Welbl et al., 2017; Gao et al., 2020), regression (Liu et al., 2017) and deep learning (Liang et al., 2018). If the final candidate selection is not based on the ranking, it may be left to the user (Nikolova, 2009), or done randomly (Araki et al., 2016; Gutl et al., 2011).

While automatic distractor generation has been widely explored for vocabulary exercises, distractors for grammar exercises have received less attention. With closed class grammatical constructions such as prepositions, many of the approaches used for vocabulary distractors are applicable. However, this greatly underrates the importance of linking distractors to the pedagogical learning goal as good distractors characterize the space of options that a learner needs to weigh against each other. Since the focus of form-based grammar exercises is not on semantics but on form, they usually rely on

ungrammatical distractors (Volodina et al., 2014). Goto et al. (2010) illustrate that for closed class target answers, the initial candidate pool consists of all types belonging to the class, whereas for open class target answers, transformations may produce suitable distractors. For the closed class of prepositions, Lee et al. (2016) start with the defined set of prepositions as candidates. For ranking, they consider co-occurrence of the candidates with either the prepositional object or the head, and their frequency as annotated errors or learner-corrected tokens in a learner corpus. Suitable for open class types, Chen et al. (2006) use distractor generation rules for a defined set of construct patterns which introduce modifications of the target answer such as morphological or syntactic variants. Aldabe et al. (2007) present an approach to generate morphological transformations of the target answer as distractor candidates and filter out those whose morpho-syntactic pattern can be found in a corpus. For verb exercises, Aldabe et al. (2009) filter the verbs from the Academic Word List by transitivity, tense and person, and rank them according to semantic similarity and distributional data. Heck and Meurers (2022) apply NLP- as well as rule-based transformations to generate well- and ill-formed variations of the target answer.

Lee et al. (2016) found distractor generation based on learner errors to yield the most plausible distractors. While their approach is closest to what we suggest, it relies on a manually annotated corpus. The resulting, statically determined distractors may be sufficiently representative for the learner population that provided the error corpus, yet they are likely to be unsuitable when more widely applied and do not allow to adapt to an individual learner’s abilities. We therefore illustrate how automatic annotations obtained from a system’s error diagnosis mechanism can effectively be used to generate and dynamically select valid and plausible distractors.

### 3 Approach

We evaluated a dataset of learner answers to form-based grammar exercises with the aim of answering the following research questions:

- RQ.1 Can the creation of learning goals, distractors and JS chunks be automated through learning analytics?
- RQ.2 Does human perception of relevant misconceptions align with relevant misconceptions

derived from learning analytics?

- RQ.3 Do errors made in half-open exercises constitute plausible distractors of closed exercises?

In order to answer RQ.1, in the following we indicate which steps of the evaluations could not be based on automated processing of the data but instead required manual labour. In addition, we determined the ability of the system’s error diagnosis module to identify relevant errors automatically. This on the one hand outlines the status quo of possible automatization and on the other hand indicates future directions for extending the module in order to support the envisioned learning analytics based adaptivity.

In order to answer RQ.2, we first identified the most frequent errors made in half-open exercises. To this end, we determined misconceptions of interest by freely annotating the entire dataset once without any reference set of potential labels. Of the thus compiled labels, those specific to questions in the simple past were included in the final label set. In order to develop an annotated learner corpus from the learner answers, we relied on two sources: (a) automatic annotations provided by the system’s error diagnosis module, and (b) manual annotations. The automatic annotations provide the single most relevant error for each learner answer. They were refined into more fine-grained labels if simple string matching was sufficient and mapped to the label set. We used these annotations whenever available ( $n = 1,778$ ) and manually annotated the remaining learner answers ( $n_{answers} = 3,058$ ,  $n_{labels} = 6,576$ ) if the system could not diagnose the nature of the error. Five annotators with backgrounds in computational linguistics annotated the learner answers independently with an unconstrained number of labels. Inter-Annotator Agreement (IAA) for the multi-label annotations of all annotators was calculated as Krippendorff’s alpha at  $\alpha = .2075$ . For the evaluations, the union set of manual and automatic annotations was used in order to not miss any potential errors. Although this might introduce some noise, it serves the purpose of identifying distractor candidates best.

In a second step, we contrasted the learner errors against misconceptions judged relevant by human exercise creators. To this purpose, we analyzed the available exercises, distractors and JS chunks of with respect to the errors for which they provide opportunities. We annotated the exercises with the

same labels used for learner error annotations. A label was assigned if it is in principle possible to make the associated error in the exercise.

Errors made in half-open exercises can only inform distractor generation if learners tend to choose the associated distractors in SC exercises. Similarly, separating constituents into individual chunks only supports learning if learners fail to put these chunks into the correct order in JS exercises. In order to answer RQ.3, we therefore analyzed whether the identified most frequent errors were also made in SC and JS exercises if the exercises provided opportunities to make them.

#### 4 Data

The evaluations are based on data obtained in the I4S project. The study collected data from 7th grade learners of English as a second language in German secondary schools who worked with the *FeedBook* over the course of a school year. The ILTS offers practice exercises in a task based setting with intelligent feedback provided to the learners as they work on the exercises. The subset of the data used for the pilot evaluations consists of the exercises on questions in the simple past.

The resulting dataset is based on 132 exercise items of the four exercise types illustrated in Fig-

ures 8–11 of Appendix A: 27 Jumbled Sentences (JS) whose chunks learners have to put into the correct order; 27 SC items for which learners need to select the correct option from the dropdown; 58 Fill-in-the-Blanks (FiB) items with input fields into which learners must write the target form; and 20 Short Answer (SA) items which require learners to write a sentence in response to a prompt. 10 of the FiB items present all correct forms to insert into the blanks as bags of words in the exercise instructions instead of giving lemmas in parentheses behind the blanks. As this renders them more similar to SC exercises, we treat them as such. FiB and SA exercises constitute half-open exercise types while SC and JS exercises are closed types. A total of 4,836 incorrect learner answers to an actionable element of the exercises was collected from 199 learners who submitted at least 1 of the exercises. An actionable element is defined as the blank of a FiB or SC exercise, a chunk of a JS exercise, or an answer to a SA exercise. All submissions were considered so that there may be multiple answers per learner and actionable element if a learner re-submitted a revised answer.

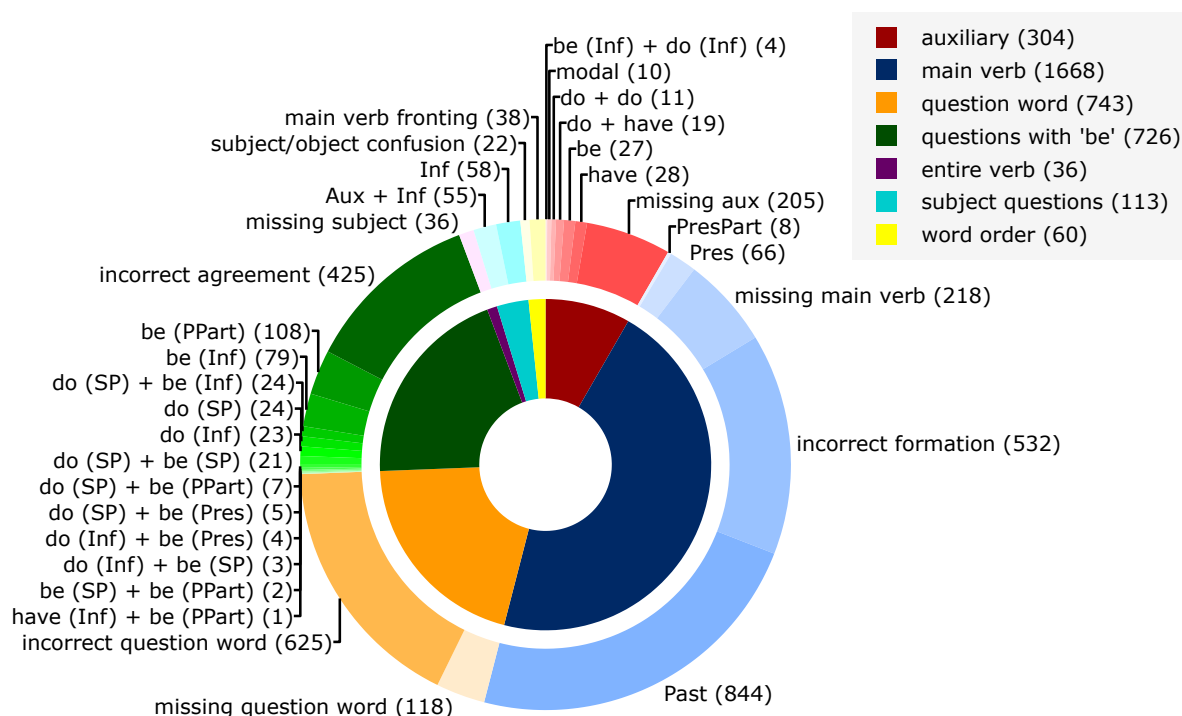


Figure 1: Frequencies of error types

## 5 Evaluation

While the focus of the analyses is on distractor generation, we also evaluated the feasibility of using the system’s error diagnosis module to determine relevant learning goals and generate chunks of JS exercises.

### 5.1 Learning goal selection

Learning goals comprise pedagogically motivated groupings of learner errors. We therefore manually identified linguistically and pedagogically related groups of error labels.

The resulting seven groups of errors that constitute important learning goals are illustrated in Figure 1: Auxiliary errors, main verb errors, errors targeting the entire verb, question word errors, word order errors, errors in questions with ‘be’, and errors in subject questions. The latter two constitute interesting special cases since question formation rules for them differ from the general rule. Their relevance as separate learning goals is strikingly emphasized when normalizing the error frequencies by the opportunities to make the respective error, as illustrated in Figure 2. Exercise generation should thus ensure to generate exercises targeting these seven learning goals for questions in the simple past.

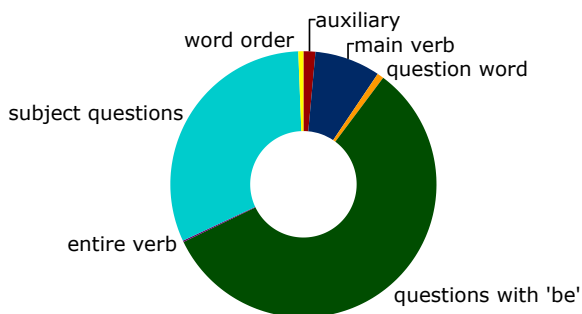


Figure 2: Normalized frequencies of error types

Focusing on RQ.1, we verified how well the system’s existing error diagnoses reflect the labels identified as relevant to exercises on questions in the simple past. To this purpose, we determined the overlap between the labels used in manual and in automatic annotations. In addition, we manually annotated a subset ( $n = 491$ ) of the automatically annotated learner errors and calculated multi-label IAA between the automatic and the joint manual annotations.

The automatic annotations cover 34 of the 63 labels found relevant for exercises on questions in the simple past. Although this leaves substantial potential

for extensions of the error diagnosis module, it also provides a solid starting point for further analyses. Automatic annotations include only a single label per error, yet IAA with the manual annotations was even slightly higher than that for the human annotators at  $\alpha = .2175$ . The error diagnosis module can therefore be used for purposes of automatic exercise generation, although both applications would benefit from extending the coverage of diagnosed learner errors.

In order to address RQ.2, we examined the exercises in the system. They evidently provide practice opportunities for all identified misconceptions as the errors were observed in the ILTS’ learner records. Yet the numbers of opportunities might differ from one misconception and exercise type to the other. In order to evaluate the available exercises’ coverage of the identified learning goals, we determined the exercise annotations’ coverage of the error labels.

The analysis reveals that not all exercise types offer practice for all misconceptions. Figure 3 illustrates that not all learning goals relevant according to the learner records can currently be practiced both with closed and half-open exercises. Thus, there is no perfect overlap between learning goals introduced by human exercise creators and those identified through learning analytics.

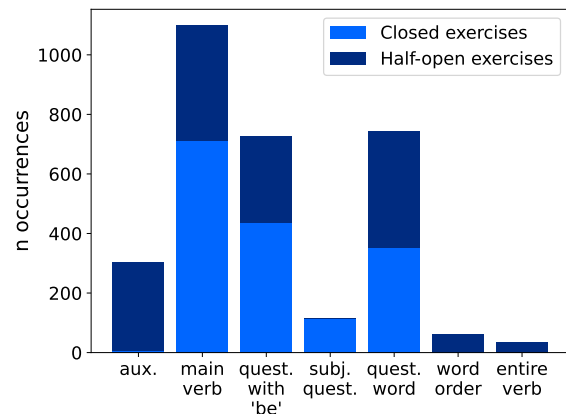


Figure 3: Error frequencies per exercise type

### 5.2 Distractor generation

While feedback generation aims to cover as many learner errors as possible, distractor generation needs to focus on the most frequent learner errors. This requires to filter the output of the answer hypotheses generated for feedback provision. Tversky (1964) found 3-option SC exercises to be



the most reliable. We therefore aimed to determine the two most frequent errors made in half-open exercises as distractors for SC exercises. Since not all error types can be made in all exercises, we normalized the occurrences of misconceptions by the number of exercise items that provided opportunities to make the error.

Figures 4–6 present normalized error frequencies per exercise type, indicating (through coloured dots next to the frequency bars) whether the system provides exercises with opportunities to make the error.

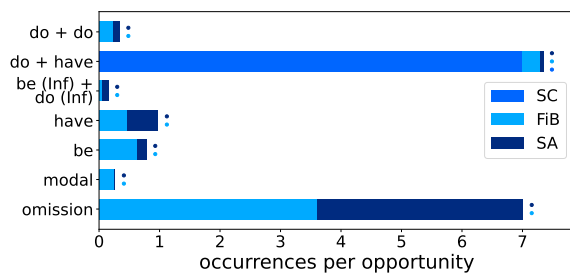


Figure 4: Frequencies of errors targeting the auxiliary

The most frequent error with respect to **auxiliaries** made by learners (see Figure 4) consists in leaving it out (e.g., Example 1a). Of the remaining errors observed in half-open exercises, using *be* (e.g., Example 1b) or *have* (e.g., Example 1c) instead of the auxiliary *do* are most frequent. Combinations of multiple auxiliaries (e.g., Example 1d) are also observed, but only in occasional submissions of half-open exercises.

- (1) What did Mr. Connor bake?
- \*What **baked** Mr. Connor?
  - \*What **was** Mr. Connor bake?
  - \*What **had** bake Mr. Connor?
  - \*What **does** Mr. Connor **have** bake?

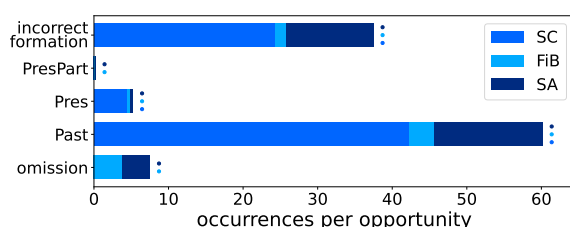


Figure 5: Frequencies of errors targeting the main verb

With respect to the **main verb** (see Figure 5), the most frequent learner error consists in using the

simple past or past participle form instead of the infinitive (e.g., Example 2a). Omitting the main verb altogether (e.g., Example 2b), using simple present – identifiable through the third person singular ‘s’ – (e.g., Example 2c), or incorrectly forming the main verb (e.g., Example 2d) were also observed rather frequently in learner answers. The latter error constitutes a special case in that it occurs only in combination with other misconceptions. Since infinitives do not transform the verb, learners always give the correct form if they intend to provide the verb in this mood. Other misconceptions appear only occasionally in learner answers.

- (2) Did you enjoy them?
- \*Did you **enjoyed** them?
  - \*Did you them?
  - \*Did you **enjoys** them?
  - \*Did you **enjoyd** them?

Only a single misconception, omitting the subject, is relevant to the learning goal practicing the **entire verb**. This error can be found with FiB as well as SA exercises, which constitute the two exercise types providing opportunities for the error.

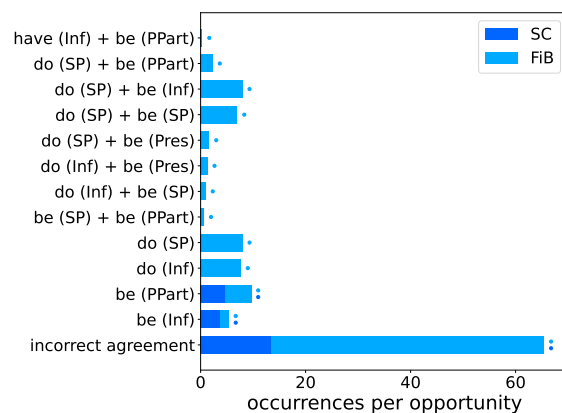


Figure 6: Frequencies of errors in questions with ‘be’

The most frequent error concerning **questions with ‘be’** (see Figure 6) by far constitutes incorrect agreement with the person of the subject (e.g., Example 3a). With FiB exercises, additional do-support (*did be*, e.g., Example 3b), *did was/were* (e.g., Example 3c), *did* (e.g., Example 3d), and *do* (e.g., Example 3e) are also frequent and should therefore be considered for distractor generation.

- (3) Were you scared?
- \***Was** you scared?
  - \***Did** you **be** scared?

- c. \***Did** you **was** scared?
- d. \***Did** you scared?
- e. \***Do** you scared?

As there are no half-open exercises available for **subject questions**, it is not possible to determine from the data what kind of errors learners would produce on their own. Observed misconceptions are therefore restricted to those offered by the SC distractors. They consist in using only the infinitive of the main verb (e.g., Example 4a) or else the infinitive with do-support (e.g., Example 4b).

- (4) Who persuaded you to come to the party?
  - a. \*Who **persuade** you to come to the party?
  - b. \*Who **did persuade** you to come to the party?

Exercises on **question words** constitute a special case in that misconceptions are specific to the target question word. The bar chart in Figure 7 illustrates that although almost all question word confusions are present in the dataset, there are clearly discernable, predominant misconceptions in the use of question words. These are, however, not bidirectional. While *where* is often incorrectly substituted by *what* or *when* in the normalized dataset, the most frequently used question words instead of *what* are *how* and *which*, and omitting the question word altogether or using *where* is the most frequent error with *when*. Instead of *why*, learners most often used *how* or *who*, whereas the most frequent question word instead of *how* is *what* or sometimes *why* in the dataset.

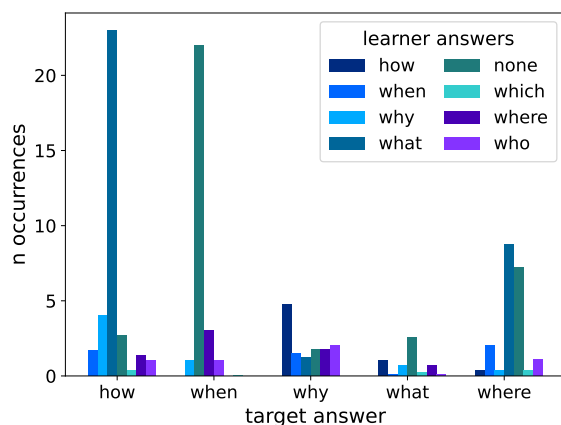


Figure 7: Frequencies of question word confusions

Turning to RQ.3, we analyzed whether the identified most frequent errors of half-open exercises appear in SC exercises as well if according distrac-

tors are available.

The system’s distractors do not cover the most frequent errors for all learning goals. With respect to the main verb, they support three of the most frequent misconceptions identified in half-open exercises: using a past form, simple present, or an incorrectly formed variant of the main verb. These distractors are selected frequently by learners in SC exercises. Concerning questions with ‘be’, SC exercises offer distractors targeting the most frequent misconception. These distractors are also selected frequently by learners. This indicates that errors observed in half-open exercises constitute plausible distractors for SC exercises, although available distractor coverage is too scarce to confirm general validity of this assumption.

With respect to RQ.1, we determined whether the automatic annotations support the labels of the two most frequent errors in half-open exercises. Looking at the individual learning goals, the most frequent misconceptions with auxiliaries – omission and the use of *be* – are both supported by the automatic annotations so that the error diagnosis module is already able to generate such distractors. The automatic annotations do not yet cover any of the misconceptions concerning the main verb, the entire verb, or questions with ‘be’. They do, however, support all labels for question word errors, thus providing the means to automatically generate according distractors.

Focusing on RQ.2, we compared the distractors introduced by human exercise creators with the most frequent errors in the learner records.

For **auxiliaries**, available distractors cover only the use of *do + have* out of the identified misconceptions. Although this distractor was selected very frequently in SC exercises, the error appears only occasionally in half-open exercises. The most frequent error with respect to this learning goal, leaving it out, is not covered by any of the SC distractors in the system. This makes sense considering that according SC exercises focusing only on the auxiliary would require an empty distractor. This exercise type thus does not lend itself well for practice of omission errors. However, the system’s SC exercises do not cover any of the remaining observed misconceptions either.

The distractors cover three of the most frequent misconceptions identified in half-open exercises practicing the **main verb**: using a past form, simple present, or incorrectly formed variant of the

main verb. Only omitting the main verb altogether, which was also observed rather frequently, is not covered for the above mentioned reason.

The system does not provide any SC exercises to practice the **entire verb**.

Concerning **questions with 'be'**, SC exercises offer distractors targeting the most frequent misconception, incorrect agreement, as well as the use of the past participle (*been*, e.g., Example 5a), and of the infinitive of *be* (e.g., Example 5b) instead of its simple past form. However, the latter two are not among the most frequent learner errors.

(5) Were you scared?

- a. \***Been** you scared?
- b. \***Be** you scared?

In general, while the distractors do not cover all misconceptions found in the learner submissions, coverage of the identified most frequent errors is high. Only those targeting word order, which is better practiced with JS exercises, and omission errors are not covered by the distractors. Concerning their pedagogical validity, solely misconceptions that are only covered by SC but not half-open exercises, i.e., those of subject questions, do not appear at all with half-open exercises. The same holds for co-occurrences of labels, indicating that the available distractors only integrate combinations of misconceptions that learners also tend to make jointly in production exercises. The manually created distractors therefore seem to be pedagogically valid since the system does not expose learners to misconceptions they would not develop of their own accord. However, in addition to the misconceptions covered by the error labels, the distractors encompass errors that have not been identified as pedagogically relevant in the manual annotation and selection process. Although both distractor creation and learning goal identification constituted manual processes, they thus put different foci on targeted misconceptions. This might indicate that exercise creators do not intuitively choose distractors that are relevant to the learning goal.

In order to compare manually created distractors to those informed by learning analytics in terms of plausibility, we followed [Haladyna and Downing \(1993\)](#)'s approach which states that at least 5% of all incorrect answers to the question need to correspond to a distractor in order for it to be plausible. We calculated the ratio of  $n$  times a distractor was selected over  $m$  times any of the item's incorrect

options was selected. Distractors obtaining a ratio lower than .05 are thus considered implausible. The evaluation shows that all distractors were selected at least once, although with differing frequencies. Only two instances of distractors were beneath the 5% threshold. While the incorrect form *forgot* may indeed be implausible, there is no clear indication as to why the form *been* was selected so rarely in the distractor group *be - was - been*, which appears in the same constellation in various other (preceding and succeeding) items, where this distractor was selected more frequently.

### 5.3 Sentence chunking

Jumbled Sentences are a natural choice of exercise type for controlled practice of word order. In order to constitute useful practice material, the chunks should fulfill two criteria: (a) They should be small enough to separate the challenging constituents that learners may struggle to assemble in the correct order. (b) On the other hand, the chunks should only be as small as necessary so as not to distract from the learning goal. We therefore analyzed word order errors with the goal of identifying constituents that should be extracted into individual chunks.

The errors particular to questions in the simple past and targeting word order concern fronting of the main verb before the subject (e.g., Example 6a), as well as interchanging the subject and the object of the sentence (e.g., Example 6b). Relevant chunks for JS exercises therefore comprise a chunk for the main verb, for the subject, and for the object.

(6) Did Mr. Jones see a doctor?

- a. \*Did **see Mr. Jones** a doctor?
- b. \*Did **a doctor** see **Mr. Jones**?

With respect to [RQ.1](#), the automatic annotations do not further distinguish between word order errors. Thus, the current error diagnosis cannot determine the most appropriate chunking for a learner.

Addressing [RQ.2](#), we analyzed the JS exercises in the system. For the first criterion concerning sentence chunking, we determined whether the exercises provide opportunities to make the word order errors observed in half-open exercises. In the exercises, 10 out of the 27 items merge the main verb with the succeeding token, thus not supporting main verb fronting errors. Only 11 items have individual chunks for the subject and the object, while the remaining 16 items have either no object or merge it with the preceding preposition or suc-

ceeding main verb.

For the second criterion, we determined the number of remaining chunks not corresponding to a constituent involved in any of the errors. To this end, we subtracted the general number of word order relevant constituents from the number of the exercise item's chunks. Allowing for some preceding and succeeding co-text, we defined results greater than two as indicative of excessive chunking. The sentences in the system are split into a mean of 5.33 chunks ( $\sigma = .88$ ) so that according to the criterion of  $n(= 3)$  relevant chunks +2, the overall number of chunks is only slightly higher than the optimal number. Considering that most exercise items merge some of the relevant chunks with preceding or succeeding tokens or do not incorporate them at all, however, the exercises do contain substantial excessive chunking.

Regarding **RQ.3**, the learner error data reveals that while JS exercises offer potential for all observed relevant word order errors, none of the learners made any main verb fronting errors in these exercises, indicating that this is only an issue in more open exercises. Subject/object confusion, on the other hand, was only observed with JS and FiB, but not with SA exercises, although all three exercise types offer opportunities for this error. Since it is of a more semantic nature, this could suggest that learners do not put much effort into semantically parsing sentences in less open-ended exercise types, rendering subject/object errors careless mistakes rather than misconceptions. Thus, neither subject/object confusion nor main verb fronting seem to be relevant for JS exercises. This might suggest that JS exercises are not relevant for practicing question formation and that word order issues arise mostly in combination with formation issues so that learners cannot practice these issues with form-controlling JS exercises. On the other hand, the fact that learners only make the errors in exercises where they have to focus on multiple linguistic aspects at once could also indicate that they lack proceduralization which would allow them to overcome processing overload. In this case, JS exercises could provide opportunities to practice each aspect in isolation.

## 6 Conclusion

We outlined a data-driven approach to determine relevant learning goals, distractors and sentence chunking for the generation of form-based gram-

mar exercises.

Addressing our first research question, we demonstrated the feasibility of using a system's error diagnosis mechanism to automatically annotate learner errors made in half-open exercises in order to dynamically adapt distractors to a learner's misconceptions. Although not all of the most frequent errors are automatically annotated in the piloted system, it is possible to extend the error diagnosis module to generate all relevant answer hypotheses. Distractor generation and error diagnosis can work hand in hand to this end. We also highlight the relevance of human involvement in the selection of pedagogically valid misconceptions. Pre-filtering of distractor templates should be manual and pedagogically motivated, while ranking of the candidates is best informed by learning analytics. The presented evaluations of most frequent learner errors based on the entire learner corpus serve as exemplary application to an adaptation module, and at the same time may be used as initial settings while the system still lacks learner records for individually adapted exercise configurations.

With respect to the second research question, we found that while there is substantial overlap between human intuition and learning analytics based exercise generation, they also differ in the focus they put on different misconceptions. Since this focus is inconsistent in human output depending on the specific task at hand, human exercise creators might benefit from explicitly specifying the learning goal in a first step. Our evaluations suggest that highest pedagogic validity of exercises can be achieved by relying on human effort to define learning goals, and on learning analytics based, automatic processing for exercise generation.

The third research question cannot be answered conclusively since the exercises do not cover all potential misconceptions for all exercise types. Where no learner data from half-open exercises is available, no conclusions can be drawn about the pedagogical validity of learner errors as distractors. This constitutes a limitation of the presented evaluations. Future work will therefore need to determine whether the errors that learners make in half-open exercises are also good distractors for SC exercises or whether learners instantly perceive them as incorrect when contrasted against the correct option. It is also yet unclear to what extent the most frequent misconceptions differ between and within learners over extended periods of time.

## References

- Itziar Aldabe and Montse Maritxalar. 2014. [Semantic Similarity Measures for the Generation of Science Tests in Basque](#). *IEEE Transactions on Learning Technologies*, 7(4):375–387.
- Itziar Aldabe, Montse Maritxalar, and Edurne Martinez. 2007. Evaluating and improving the distractor-generating heuristics. In *Workshop on NLP for Educational Resources*. In conjunction with RANLP07, pages 7–13.
- Itziar Aldabe, Montse Maritxalar, and Ruslan Mitkov. 2009. [A Study on the Automatic Selection of Candidate Sentences Distractors](#). In *Artificial Intelligence in Education*, volume 200, pages 656–658.
- Tahani Alsubait, Bijan Parsia, and Uli Sattler. 2015. [Generating Multiple Choice Questions From Ontologies: How Far Can We Go?](#) In *Knowledge Engineering and Knowledge Management: EKAW 2014 Satellite Events, VISUAL, EKMI, and ARCOE-Logic, Linköping, Sweden, November 24-28, 2014. Revised Selected Papers. 19*, pages 66–79. Springer.
- Jun Araki, Dheeraj Rajagopal, Sreecharan Sankaranarayanan, Susan Holm, Yukari Yamakawa, and Teruko Mitamura. 2016. Generating Questions and Multiple-Choice Answers using Semantic Analysis of Texts. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1125–1136.
- Carmen Becker and Jana Roos. 2016. [An approach to creative speaking activities in the young learners’ classroom](#). *Education Inquiry*, 7(1):27613.
- Jonathan Brown, Gwen Frishkoff, and Maxine Eskenazi. 2005. [Automatic Question Generation for Vocabulary Assessment](#). In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, HLT ’05*, pages 819–826, USA. Association for Computational Linguistics.
- Chia-Yin Chen, Hsien-Chin Liou, and Jason S. Chang. 2006. [FAST – An Automatic Generation System for Grammar Tests](#). In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 1–4.
- Tao Chen, Naijia Zheng, Yue Zhao, Muthu Kumar Chandrasekaran, and Min-Yen Kan. 2015. [Interactive Second Language Learning from News Websites](#). In *Proceedings of the 2nd Workshop on Natural Language Processing Techniques for Educational Applications*, pages 34–42.
- David Coniam. 1997. [A Preliminary Inquiry Into Using Corpus Word Frequency Data in the Automatic Generation of English Language Cloze Tests](#). *CALICO Journal*, 14.
- Rui Correia, Jorge Baptista, Maxine Eskenazi, and Nuno J. Mamede. 2012. [Automatic Generation of Cloze Question Stems](#). In *PROPOR*, pages 168–178. Springer.
- Lingyu Gao, Kevin Gimpel, and Arnar Jensson. 2020. [Distractor Analysis and Selection for Multiple-Choice Cloze Questions for Second-Language Learners](#). In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 102–114.
- Donna Marie Gates. 2011. [How to Generate Cloze Questions from Definitions: a Syntactic Approach](#). In *2011 AAAI Fall symposium series*.
- Takuya Goto, Tomoko Kojiri, Toyohide Watanabe, Tomoharu Iwata, and Takeshi Yamada. 2010. [Automatic Generation System of Multiple-Choice Cloze Questions and its Evaluation](#). *Knowledge Management & E-Learning: An International Journal*, 2:210–224.
- Qi Guo, Chinmay Kulkarni, Aniket Kittur, Jeffrey P. Bigham, and Emma Brunskill. 2016. [Questimator: Generating Knowledge Assessments for Arbitrary Topics](#). In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI’16*, pages 3726–3732, New York, New York, USA. AAAI Press.
- Christian Gutl, Klaus Lankmayr, Joachim Weinhofer, and Margit Hofler. 2011. [Enhanced Automatic Question Creator–EAQC: Concept, Development and Evaluation of an Automatic Test Item Creation Tool to Foster Modern e-Education](#). *Electronic Journal of e-Learning*, 9(1):23–38.
- Thomas M. Haladyna and Steven M. Downing. 1993. [How Many Options is Enough for a Multiple-Choice Test Item?](#) *Educational and Psychological Measurement*, 53(4):999–1010.
- Tanja Heck and Detmar Meurers. 2022. [Parametrizable exercise generation from authentic texts: Effectively targeting the language means on the curriculum](#). In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 154–166, Seattle, Washington. Association for Computational Linguistics.
- Jennifer Hill and Rahul Simha. 2016. [Automatic generation of context-based fill-in-the-blank exercises using co-occurrence likelihoods and Google n-grams](#). In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 23–30.
- Shu Jiang and John Lee. 2017. [Distractor Generation for Chinese Fill-in-the-blank Items](#). In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 143–148, Copenhagen, Denmark. Association for Computational Linguistics.
- Nikiforos Karamanis, Ruslan Mitkov, et al. 2006. [Generating Multiple-Choice Test Items from Medical Text: A Pilot Study](#). In *Proceedings of the fourth international natural language generation conference*, pages 111–113.

- Susanne Knoop and Sabrina Wilske. 2013. WordGap - Automatic Generation of Gap-Filling Vocabulary Exercises for Mobile Learning. In *Proceedings of the second workshop on NLP for computer-assisted language learning at NODALIDA 2013; May 22-24; Oslo; Norway. NEALT Proceedings Series 17*, 086, pages 39–47. Citeseer.
- Girish Kumar, Rafael Banchs, and Luis D’Haro. 2015. RevUP: Automatic Gap-Fill Question Generation from Educational Texts. In *BEA@NAACL-HLT*, pages 154–161.
- John Lee and Stephanie Seneff. 2007. Automatic Generation of Cloze Items for Prepositions. In *Eighth Annual Conference of the International Speech Communication Association*.
- John Lee, Donald Sturgeon, and Mengqi Luo. 2016. A CALL System for Learning Preposition Usage. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 984–993, Berlin, Germany. Association for Computational Linguistics.
- Chen Liang, Xiao Yang, Neisarg Dave, Drew Wham, Bart Pursel, and C. Lee Giles. 2018. Distractor Generation for Multiple Choice Questions Using Learning to Rank. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 284–290, New Orleans, Louisiana. Association for Computational Linguistics.
- Chen Liang, Xiao Yang, Drew Wham, Bart Pursel, Rebecca Passonneau, and C. Lee Giles. 2017. Distractor Generation with Generative Adversarial Nets for Automatically Creating Fill-in-the-Blank Questions. In *Proceedings of the Knowledge Capture Conference, K-CAP 2017*, New York, NY, USA. Association for Computing Machinery.
- Chao-Lin Liu, Chun-Hung Wang, Zhao Ming Gao, and Shang-Ming Huang. 2005. Applications of Lexical Information for Algorithmically Composing Multiple-Choice Cloze Items. In *Proceedings of the second workshop on Building Educational Applications Using NLP*, pages 1–8.
- Ming Liu, Vasile Rus, and Li Liu. 2017. Automatic Chinese Multiple Choice Question Generation Using Mixed Similarity Strategy. *IEEE Transactions on Learning Technologies*, 11(2):193–202.
- Montse Maritxalar, Elaine Dhonnchadha, Jennifer Foster, and Monica Ward. 2011. Quizzes on Tap: Exporting a Test Generation System from One Less-Resourced Language to Another. In *Human Language Technology Challenges for Computer Science and Linguistics*, volume 8387, pages 502–514, Cham. Springer International Publishing.
- Detmar Meurers. 2012. Natural Language Processing and Language Learning. In *The Encyclopedia of Applied Linguistics*. John Wiley & Sons, Ltd.
- Detmar Meurers, Ramon Ziai, Luiz Amaral, Adriane Boyd, Aleksandar Dimitrov, Vanessa Metcalf, and Niels Ott. 2010. Enhancing Authentic Web Pages for Language Learners. In *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 10–18, Los Angeles, California. Association for Computational Linguistics.
- 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.
- Ruslan Mitkov, Andrea Varga, Luz Rello, et al. 2009. Semantic similarity of distractors in multiple-choice tests: extrinsic evaluation. In *Proceedings of the workshop on geometrical models of natural language semantics*, pages 49–56.
- Josef Robert Moser, Christian Gütl, and Wei Liu. 2012. Refined Distractor Generation with LSA and Styliometry for Automated Multiple Choice Question Generation. In *Australasian Conference on Artificial Intelligence*, pages 95–106. Springer.
- Jack Mostow and Hyeju Jang. 2012. Generating Diagnostic Multiple Choice Comprehension Cloze Questions. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 136–146.
- Ivelina Nikolova. 2009. New Issues and Solutions in Computer-aided Design of MCTI and Distractor Selection for Bulgarian. In *Proceedings of the Workshop Multilingual resources, technologies and evaluation for central and Eastern European languages*, pages 40–46.
- Andreas Papasalouros, Konstantinos Kanaris, and Konstantinos I. Kotis. 2008. Automatic Generation Of Multiple Choice Questions From Domain Ontologies. In *e-Learning*.
- Cora Parrisius, Ines Pieronczyk, Carolyn Blume, Katharina Wendebourg, Diana Pili-Moss, Mirjam Assmann, Sabine Beilharz, Stephen Bodnar, Leona Colling, Heiko Holz, et al. 2022a. Using an Intelligent Tutoring System within a Task-Based Learning Approach in English as a Foreign Language Classes to Foster Motivation and Learning Outcome (Interact4School): Pre-registration of the Study Design.
- Cora Parrisius, Katharina Wendebourg, Sven Rieger, Ines Loll, Diana Pili-Moss, Leona Colling, Carolyn Blume, Ines Pieronczyk, Heiko Holz, Stephen Bodnar, et al. 2022b. Effective Features of Feedback in an Intelligent Tutoring System-A Randomized Controlled Field Trial (Pre-Registration).
- Rajkumar Patil, Sachin Bhaskar Palve, Kamesh Vell, and Abhijit Vinod Boratne. 2016. Evaluation of multiple choice questions by item analysis in a medical college at Pondicherry, India. *International Journal of Community Medicine and Public Health*, 3(6):1612–1616.

- Naiara Pérez and Montse Cuadros. 2017. [Multilingual CALL Framework for Automatic Language Exercise Generation from Free Text](#). In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 49–52, Valencia, Spain. Association for Computational Linguistics.
- Ildikó Pilán and Elena Volodina. 2014. [Reusing Swedish FrameNet for training semantic roles](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1359–1363, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Juan Pino, Michael Heilman, and Maxine Eskenazi. 2008. A Selection Strategy to Improve Cloze Question Quality. In *Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains. 9th International Conference on Intelligent Tutoring Systems, Montreal, Canada*, pages 22–32.
- Pei Quan, Yong Shi, Lingfeng Niu, Ying Liu, and Tianlin Zhang. 2018. [Automatic Chinese multiple-choice question generation for human resource performance appraisal](#). *Procedia Computer Science*, 139:165–172.
- Björn Rudzewitz, Ramon Ziai, Kordula De Kuthy, Verena Möller, Florian Nuxoll, and Detmar Meurers. 2018. [Generating Feedback for English Foreign Language Exercises](#). In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 127–136, New Orleans, Louisiana. Association for Computational Linguistics.
- Keisuke Sakaguchi, Yuki Arase, and Mamoru Komachi. 2013. [Discriminative Approach to Fill-in-the-Blank Quiz Generation for Language Learners](#). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 238–242.
- Karim Shabani, Khatib Mohammad, and Saman Ebadi. 2010. [Vygotsky's Zone of Proximal Development: Instructional Implications and Teachers' Professional Development](#). *English Language Teaching*, 3.
- Chi-Chiang Shei. 2001. [FollowYou!: An Automatic Language Lesson Generation System](#). *Computer Assisted Language Learning*, 14(2):129–144.
- Simon Smith, P. V. S. Avinesh, and Adam Kilgarriff. 2010. Gap-fill Tests for Language Learners: Corpus-Driven Item Generation. In *Proceedings of ICON-2010: 8th International Conference on Natural Language Processing*, pages 1–6. Macmillan Publishers India.
- Simon Smith, Adam Kilgarriff, Gong Wen-liang, Scott Sommers, and Wu Guang-zhong. 2009. Automatic Cloze Generation for English Proficiency Testing. In *Proceedings of the LITC Conference*.
- Nina Spada and Yasuyo Tomita. 2010. [Interactions Between Type of Instruction and Type of Language Feature: A Meta-Analysis](#). *Language Learning*, 60(2):263–308.
- Katherine Stasaski and Marti A. Hearst. 2017. [Multiple Choice Question Generation Utilizing An Ontology](#). In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 303–312.
- Eiichiro Sumita, Fumiaki Sugaya, and Seiichi Yamamoto. 2005. [Measuring Non-native Speakers' Proficiency of English by Using a Test with Automatically-Generated Fill-in-the-Blank Questions](#). In *Proceedings of the second workshop on Building Educational Applications Using NLP*, pages 61–68.
- Yuni Susanti, Ryu Iida, and Takenobu Tokunaga. 2015. [Automatic Generation of English Vocabulary Tests](#). In *CSEUDU (1)*, pages 77–87.
- Yuni Susanti, Takenobu Tokunaga, Hitoshi Nishikawa, and Hiroyuki Obari. 2018. [Automatic distractor generation for multiple-choice English vocabulary questions](#). *Research and Practice in Technology Enhanced Learning*, 13(1):15.
- Amos Tversky. 1964. [On the optimal number of alternatives at a choice point](#). *Journal of Mathematical Psychology*, 1(2):386–391.
- Elena Volodina, Ildikó Pilán, Lars Borin, and Therese Lindström Tiedemann. 2014. [A flexible language learning platform based on language resources and web services](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3973–3978, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. [Crowdsourcing Multiple Choice Science Questions](#). In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106, Copenhagen, Denmark. Association for Computational Linguistics.
- Masaru Yamada. 2019. Language learners and non-professional translators as users. In Minako O'Hagan, editor, *The Routledge handbook of translation and technology*, chapter 11, pages 183–199. Routledge.
- Chak Yan Yeung, John Sie Yuen Lee, and Benjamin Ka-Yin T'sou. 2019. Difficulty-aware Distractor Generation for Gap-Fill Items. In *Australasian Language Technology Association Workshop*.
- Torsten Zesch and Oren Melamud. 2014. [Automatic Generation of Challenging Distractors Using Context-Sensitive Inference Rules](#). In *Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 143–148.

Branko Žitko, Slavomir Stankov, Marko Rosić, and Ani Grubišić. 2009. [Dynamic test generation over ontology-based knowledge representation in authoring shell](#). *Expert Systems with Applications*, 36(4):8185–8196.

## A Exercise types

Please drag the sentence parts into the correct order to form a question in the simple past.

he did When ? come home

\_\_\_\_\_

\_\_\_\_\_



 

Figure 8: Jumbled Sentences

Please select the correct option to form a question in the simple past.

When did he  (come) home?

came  
come  
comed

Figure 9: Single Choice

Please fill the gap with the form of the verb to form a question in the simple past.

When did he \_\_\_\_\_ (come) home?

Figure 10: Fill-in-the-Blanks

Please give a question in the simple past asking about the masked part of the sentence.

He came home at XXXXXX.


\_\_\_\_\_ 

Figure 11: Short Answer



# Reviewriter: AI-Generated Instructions For Peer Review Writing

Xiaotian Su<sup>1</sup>, Thiemo Wambsganss<sup>1</sup>, Roman Rietsche<sup>2</sup>,  
Seyed Parsa Neshaei<sup>1</sup>, Tanja Käser<sup>1</sup>

<sup>1</sup> EPFL, Lausanne, Switzerland

{xiaotian.su, thiemo.wambsganss, seyed.neshaei, tanja.kaeser}@epfl.ch

<sup>2</sup> Universtiy of St.Gallen, St.Gallen, Switzerland

roman.rietsche@hsg.ch

## Abstract

Large Language Models (LLMs) offer novel opportunities for educational applications that have the potential to transform traditional learning for students. Despite AI-enhanced applications having the potential to provide personalized learning experiences, more studies are needed on the design of generative AI systems and evidence for using them in real educational settings. In this paper, we design, implement and evaluate Reviewriter, a novel tool to provide students with AI-generated instructions for writing peer reviews in German. Our study identifies three key aspects: a) we provide insights into student needs when writing peer reviews with generative models which we then use to develop a novel system to provide adaptive instructions b) we fine-tune three German language models on a selected corpus of 11,925 student-written peer review texts in German and choose German-GPT2 based on quantitative measures and human evaluation, and c) we evaluate our tool with fourteen students, revealing positive technology acceptance based on quantitative measures. Additionally, the qualitative feedback presents the benefits and limitations of generative AI in peer review writing.

## 1 Introduction

Peer reviewing is a process by which learners provide formative feedback to each other on an individual task based on assessment criteria (Sadler and Good, 2006; Rietsche and Söllner, 2019). Research has found theoretical and empirical evidence for the positive effects of peer reviews on critical thinking skills (Lin et al., 2021; Ibarra-Sáiz et al., 2020), communication skills (Lai, 2016), and learning motivations (Hsia et al., 2016). The prevailing practice of peer review in tertiary education is evident in the eruption of massive open online courses (MOOCs) (Li et al., 2016). In these large-scale learning scenarios, peer review is particularly important since it is challenging for teachers to give effective one-by-one feedback due to immersive workload and

shortage of time (Er et al., 2021). However, according to Oliver (1982), a challenge that plagues many student writers, including those having satisfactory grammar and spelling skills, is writer’s block. It was defined by Rose (1980) as "that frustrating, self-defeating inability to generate the next line, the right phrase, the sentence that will release the flow of words again." A collaborator who provides instructions and points out new directions might help alleviate writer’s block (Clark et al., 2018) and the combination of a writer’s own ideas with suggested ideas is a form of psychological creativity (Boden et al., 2004). Novel LLMs have the potential to address the challenge of writer’s block by generating suggestions for the next lines, right phrases, or sentences, thereby facilitating the flow of ideas (Gero et al., 2022), and helping students compose responses more efficiently (van Dis et al., 2023; Gao and Jiang, 2021). There are LLM-based collaborative writing tools to provide support for various writing tasks, including story writing (Yang et al., 2022), science writing (Gero et al., 2022), and screenwriting (Mirowski et al., 2022). However, few have investigated the utilization of generative AI for peer review writing tasks. Therefore, in this paper, we build and evaluate Reviewriter which can provide AI-generated instructions tailored to students’ needs while writing peer reviews. It suggests possible directions based on students’ input to inspire divergent outcomes while still leaving learners in control of the final text.

To investigate how to provide students with help to overcome writer’s block in peer review writing, we conduct a literature review to gather insights for a peer review support system. We summarize five user requirements from interviews with twelve graduate students. Based on those, we develop seven design principles for providing AI-generated instructions in peer review tasks. Next, we search peer review corpora satisfying certain criteria and pre-process 11,925 student-written peer

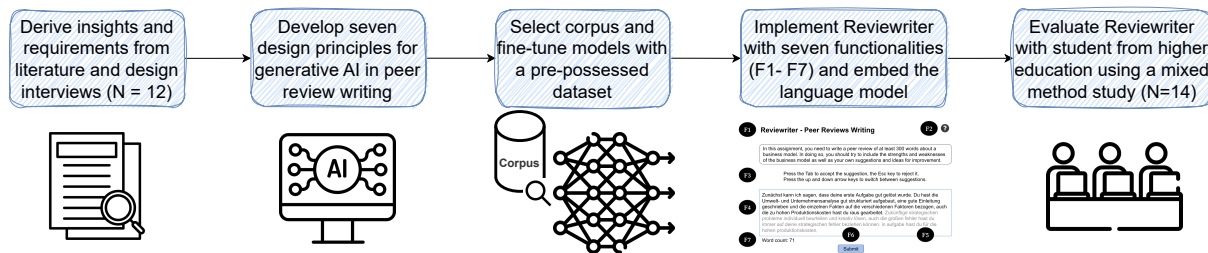


Figure 1: Overview of our methodology: We first gather system needs and requirements from literature and student interviews. Then we derive seven design principles with pedagogical considerations for a tool to provide AI-generated instructions for peer review writing tasks. Next, we fine-tuned three language models based on a selected corpus (Wambsganss et al., 2022b). Then, we instantiate the design in Reviewriter and evaluate it with fourteen students to assess its performance and gather quantitative as well as qualitative feedback.

review texts in German (Wambsganss et al., 2022b). We use it to fine-tune three language models to provide students with informative instructions. The best results according to training loss and human evaluation of fluency and correctness are achieved by German GPT-2. Then, we implement the design principles into the system to provide AI-generated instructions for peer review writing. Finally, in a mixed-method study with our full-working prototype, we evaluate the performance of the tool in a real-world learning exercise with fourteen students, and four of them also participated in the design interview. We assess the technology acceptance and level of enjoyment of the tool using well-defined constructs from Venkatesh and Bala (2008); Venkatesh et al. (2003) and also collect qualitative feedback from students.

Our research makes three contributions to the innovative use of NLP in education. Firstly, we provide insights and practical design considerations for incorporating AI-generated instructions in peer review writing tasks to overcome the known challenge of writer’s block (Oliver, 1982). Secondly, we present and compare three open-source language models fine-tuned on a selected corpus of 11,925 student-written peer review texts in German. Lastly, we build Reviewriter, which implements seven functionalities with pedagogical design considerations and evaluates it on fourteen students from tertiary education. Our findings suggest that the tool providing AI-generated instructions in students’ peer writing tasks leads to high ease of use and a high intention to use for students in their review writing process. Moreover, in the qualitative feedback, we find that the model has the potential to provide novel ideas for students to continue in depth. However, like other LLMs, it suffers

from hallucination (Maynez et al., 2020) by producing factually incorrect and nonsensical answers, this invites further research to overcome and mitigate artificial hallucination. With Reviewriter, we present an interface with design rationales and an evaluated tool that other researchers can build upon to explore the effects of LLMs and the benefits and limitations of generative AI for writing peer reviews and building educational applications.

## 2 Related work

### 2.1 Student peer reviewing

There has always been significant interest in the study of peer reviews in the NLP community. Jia et al. (2022) introduced an approach called incremental zero-shot learning (IZSL) to address the issue of insufficient historical data for peer reviews. Wambsganss et al. (2022a) used empathy detection algorithms from NLP to analyze the given text and provide adaptive feedback in students’ peer writing process. Moreover, several works have investigated how to embed classification models to support students in peer review writing. For example, researchers have explored the use of these models to develop argumentation skills (Wambsganss et al., 2020), support cognitive and emotional empathy writing (Wambsganss et al., 2021), and assess the specificity of written peer feedback (Rietsche et al., 2022). While NLP models, particularly LLMs, have the potential to deliver adaptive learning content (Adiguzel et al., 2023; Qadir, 2022), little research has focused on how to leverage their ability to provide tailored instructions for students during peer review writing (Darvishi et al., 2022). van Dis et al. (2023) mentioned benefits provided by generative AI for completing peer review tasks quickly. Experimental results from Gao and

Jiang (2021) showed that the effectiveness of generated suggestions, regardless of their performance quality, has consistently helped humans compose responses more efficiently when providing suggestions. In addition, Gero et al. (2022) demonstrated that students find it faster and easier to draw on language from generated texts than to write a sentence from scratch, even when given well-known information. Therefore, we propose a novel peer review writing tool *Reviewriter*, by leveraging the power of generative models, it can provide students with adaptive instructions to help them overcome writer’s block in peer review writing.

## 2.2 NLP for writing support

With the massive success of ChatGPT, NLP is rapidly evolving as a key tool in writing support. On one hand, there is widespread adoption of generative AI in practice. Commercial writing assistants like Monica <sup>1</sup>, a ChatGPT-powered extension, can support copywriting. And specialized applications like Jenni AI <sup>2</sup>, Jasper AI <sup>3</sup> and Notion AI <sup>4</sup> can support creative writing. They are not only able to complete sentences but also generate the whole blog post and many other types of content including essays, emails, stories, and speeches based on users’ input. On the other hand, many studies have focused on the use of language models for writing support in tertiary education. For instance, researchers have explored the use of these models for academic writing (Gero et al., 2022), fiction writing (Yang et al., 2022), and text summarization (Dang et al., 2022). Despite the widespread adoption of NLP in writing instruction, many models, including ChatGPT, remain general-purpose tools that have not been fine-tuned for specific tasks (Chen et al., 2023) or designed for particular educational settings (Kuhail et al., 2023). Embedding the AI techniques in a student-centered design is a complex task with several socio-technical challenges (Xu et al., 2021), including data collection (Zawacki-Richter et al., 2019), potential bias (Adiguzel et al., 2023) or discrimination (Pedróf et al., 2019) in the data, inadequate dataset training (Kuhail et al., 2023), incorporating the models, lack of student involvement in the design process (Verleger and Pembroke, 2018), lacking feedback on the generative system (Kuhail et al., 2023), and evaluating

<sup>1</sup><https://monica.im/>

<sup>2</sup><https://jenni.ai/>

<sup>3</sup><https://www.jasper.ai>

<sup>4</sup><https://www.notion.so/product/ai>

student perceptions (Xu et al., 2021). The present work provides insights into how to embed generative AI into peer review writing by establishing student-centered design with pedagogical considerations. We carefully select an unbiased corpus with a sufficient amount of peer review text to fine-tune language models. Furthermore, we evaluate student perceptions quantitatively and collect qualitative feedback on the generative AI system.

## 3 Generative modeling to provide students adaptive instructions

### 3.1 The peer review dataset

To make sure our system is skilled in providing adaptive instructions for writing peer reviews and to improve accuracy and efficiency for human-AI interaction (Lee et al., 2022b), we decide to fine-tune language models with a peer review dataset. We start by searching the literature for a corpus that fulfilled the following criteria: a) it contains a large amount of student-written text in one particular domain (e.g., business model feedback) (Kuhail et al., 2023), b) it consists of a sufficient size to represent different nuances of characteristics in a balanced fashion (e.g. specificity, helpfulness) (Rietsche et al., 2022), and c) it does not possess a significant bias (e.g. gender, racial or social discrimination) (Adiguzel et al., 2023). The business model peer review corpus published in Wambsganss et al. (2022b) fulfilled all these requirements. The corpus consists of 11,925 peer reviews collected at a university in the German-speaking area of Europe. They were written by first-year master’s students in a business department course. The student population has an average age of 24.6 years old with a standard deviation of 1.7 years. Students wrote approximately 9 peer reviews per course with an average length of 220 words. Furthermore, Wambsganss et al. (2022b) showed that this collected corpus does not reveal many biases in nine WEAT co-occurrence analyses or in the GloVe embeddings. This corpus provides us with a sufficient amount of unbiased peer review texts to fine-tune language models for adaptive instructions in the domain of business peer reviews.

### 3.2 Data pre-processing

To ensure the model could generate high-quality instructional text, we select reviews written from 2016 to 2021 with a rated helpfulness score greater than five on a 1 - 7 Likert Scale (1: low, 4: neutral,

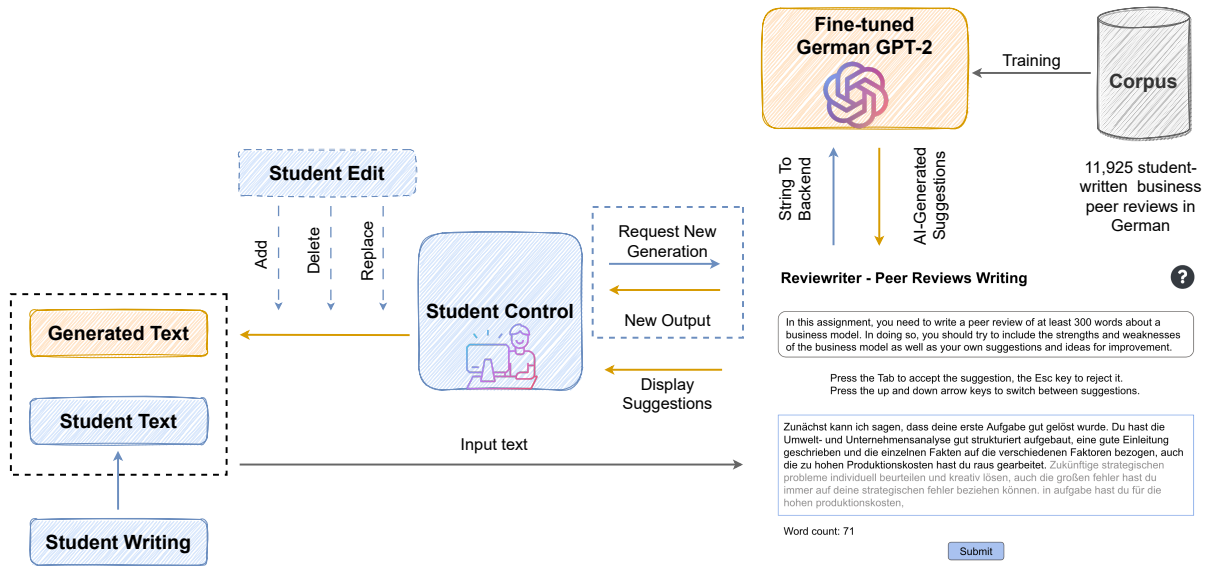


Figure 2: Architecture of Reviewriter to provide AI-generated instructions for students to write peer reviews. First, students enter initial input, which is then used by the German GPT-2 model to generate instructions. The students evaluate the generated content and decide whether to regenerate it. Following this, students are free to edit the instructions. Finally, both the generated text and the student’s text are utilized as inputs for the next generation.

7: high). We start by removing HTML tags, irrelevant information like PDF file names and specific information like URLs, keywords (revealing the identity of students), and questions asked to write reviews which some students copied to their review text (Appendix A.1). We also expand abbreviations as shown in Appendix A.2. Then, we shuffle and divide cleaned data into train and test datasets with proportions of 0.8 and 0.2 for fine-tuning and evaluating the language model. Lastly, all sentences are tokenized with model-specific tokenizers.

### 3.3 The generative models

Transformer-based language models, such as BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019), using the pre-training and fine-tuning paradigm, have revolutionized NLP and achieved state-of-the-art records on various tasks. These models are first pre-trained in a self-supervised fashion on a large corpus and fine-tuned for specific downstream tasks (Wang et al., 2018). In our case, to provide AI-generated instructions for German peer review writing, we use pre-trained causal language models on the HuggingFace platform (Wolf et al., 2020) for German text generation. We choose them because there is no usage limitation and by utilizing open-source technology, we contribute to LLM transparency (van Dis et al., 2023; Adiguzel et al., 2023), allowing other researchers to easily replicate our find-

ings or build upon them. Therefore, we selected two German GPT-2 models (dbmdz/german-gpt2<sup>5</sup> and benjamin/gerpt2-large<sup>6</sup>) and one multilingual model BLOOM (Scao et al., 2022) (bigscience/bloom-560m<sup>7</sup>). We did not use GPT3 for fine-tuning since it was not open-source available at the time of our research. For all of them, we fine-tune the pre-trained models following the default hyperparameter settings (Appendix A.3) with block size 128, and 500 warm-up steps.

We compare training loss and used human evaluation to select the best model. Note that GerPT2-large already performs well (Appendix A.4 for sample generated text) after ten epochs of training, even with higher training loss compared to the other two models (Table 1). However, it suffers a long inference time (a student needs to wait around 10 seconds to get instructions given 40 words) compared to the other two models (5 seconds with the same input). Therefore, we decide to further evaluate German GPT-2 and BLOOM. We conduct a human evaluation of the quality of the generated response. Specifically, we sample ten instructions generated by each model and present them to two German researchers to evaluate their fluency and correctness. From the evaluation of both parties, German GPT-2 yields more coherent results than

<sup>5</sup><https://huggingface.co/dbmdz/german-gpt2>

<sup>6</sup><https://huggingface.co/bigscience/bloom-560m>

<sup>7</sup><https://huggingface.co/benjamin/gerpt2-large>

the BLOOM model and there are more meaningless sentences from the response generated by BLOOM than by German GPT-2. Therefore, we decide to use the German GPT-2 model as the base for the tool with a default temperature of 1.0 for generating the next token.

PLM	Size # Param.	Training loss	Training epochs
German GPT-2	124	0.0418	30
BLOOM	560M	0.0560	30
GerPT2-large	774M	2.8183	10

Table 1: Comparison of the number of parameters for three transformer-based pretrained language models (PLMs) and their training and evaluation loss.

### 3.4 The generative system

To design a system providing AI-generated instructions for peer review writing, we first draw on insights from relevant literature. Following the methodology of Cooper (1988), we analyze human-AI interaction (Shen and Wu, 2023; Chan et al., 2023; Lee et al., 2022b) and NLP-supported peer review systems (Alqassab et al., 2023; Darvishi et al., 2022). Then, to gather insights into the needs of writing peer reviews with AI-generated instructions for tertiary education, we conduct semi-structured interviews with twelve graduate students. We reach out to a group of computer science students who previously registered in a business class and have experience writing peer reviews on business models, and to students in our university for general recruitment. The participants have a diverse background in computer science, business, or psychology, and a mean age of 24.50 years (SD = 2.02), including two females and ten males (representing the distribution of computer science students at our school). Half of them had experience writing peer reviews, while the others did not. Each interview lasts around 30 to 50 minutes. We use the expert qualitative interview method outlined in Brinkmann (2013) and Gläser and Laudel (2009) to gain an initial understanding of students’ needs for receiving adaptive instructions in peer review writing. We ask topics about prior experience with technology-based writing systems, perceptions of existing writing systems (e.g., Grammarly), difficulties in writing peer reviews, and desired functionalities for a system to support peer review writing. We transcribe the interviews and identify five

clusters of requirements following Cohn (2004). We find that 75% of the students would like to interact with a clean and straightforward interface (*user requirement - UR 1*). Two-thirds of interviewees asked for intuitive guidance on how to interact with the tool (*UR 2*). And 41.7% of them said that they would like to see more than one instruction to choose from (*UR 3*). One-third of the students stated that they prefer to view a complete piece of instruction rather than words or phrases to formulate a concrete idea (*UR 4*). Lastly, two-thirds of them indicated that they would like to see the number of words they have entered to have better control over the structure of the review (*UR 5*).

	Design Principle
DP1)	Provide a web-based application with a responsive clean and intuitive interface to allow students to use the tool with ease and stay motivated to write.
DP2)	Provide clear and detailed guidance to ensure that students understand how to use the tool and can take full advantage of the features offered.
DP3)	Provide an intuitive keyboard control to make it easy for students to manipulate the AI-generated instructions.
DP4)	Provide a simple text area for students to write, edit the peer review, and view multiple inline instructions.
DP5)	Present instructions in an inline format in the text area to help students quickly pick up ideas while allowing them to stay in the context of writing to reduce cognitive burden.
DP6)	Provide a complete argument for each instruction to assist students in constructing comprehensive reviews.
DP7)	Present a summary of statistics on the text to guide students on how many words they have written.

Table 2: Derived design principles on how to provide AI-generated instructions for students to write peer reviews.

With insights derived from the literature review and requirements from student interviews (similar to Rietsche et al. (2018)), we develop seven design principles (Table 2) and further map them to seven functionalities (Figure 3 *F1 - F7*) in Reviewriter, a responsive web application to provide AI-generated instructions for peer review writ-

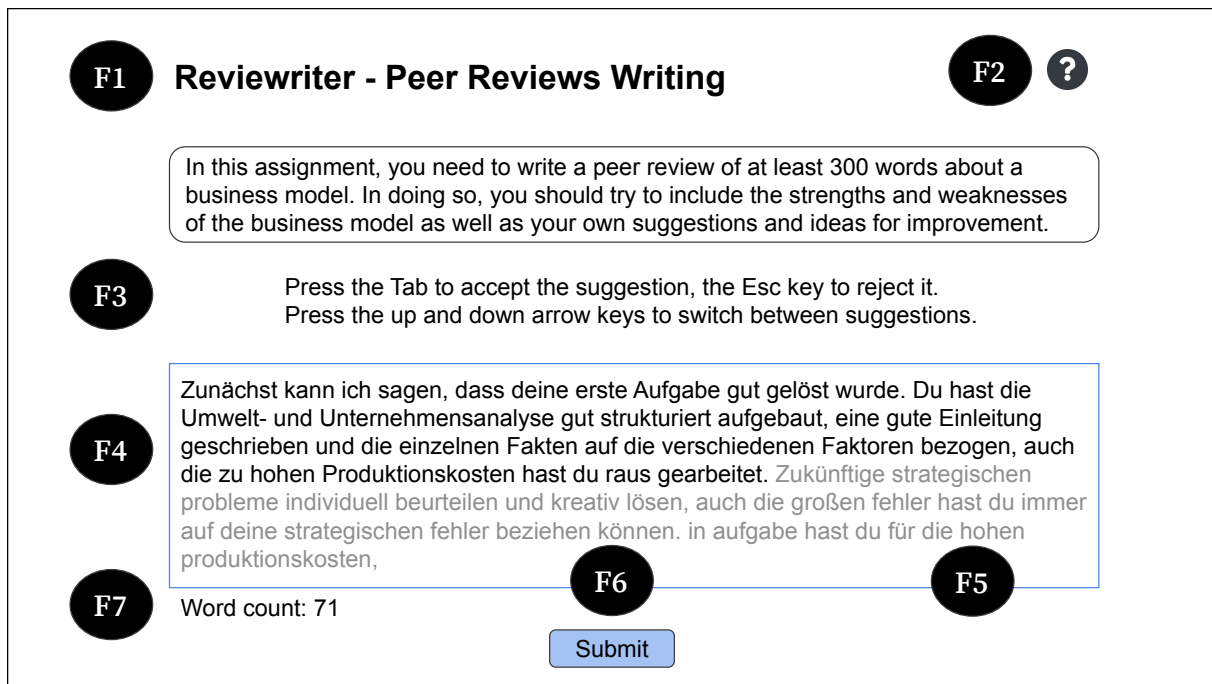


Figure 3: A screenshot of Reviewriter and its main functionalities (*F1 - F7*) derived from system requirements and design principles. The system provides a clean interface (*F1*). By clicking the question mark, students get detailed guidance on the peer review writing task and the usage of the tool (*F2*). A simple text area supports all typical interactions, such as typing, selecting, editing, and deleting text, and caret movement via keys and mouse (*F4*). In the input area, the sentences in black are the actual text, we display the AI-generated instruction in an inline format in gray (*F5*). The model generates next-sentence predictions to give students a complete view of the idea (*F6*). We provide three instructions each time, and students may use the *Tab* key to accept, the *Esc* key to reject, and the *Up* and *Down* arrow keys to toggle through different instructions (*F3*). The total number of words is displayed below the text area to inform students of their writing progress (*F7*).

ing. The design is student-centered and has two main components: a neat interface with key commands for text editing (Figure 3) and a generative language model in the backend 3.3. To foster the independent thinking of students and discourage over-reliance on technology (Adiguzel et al., 2023), we organize a workshop with two senior researchers to deliberate on the optimal timing for presenting the generated instructions. Combined with studies Buschek et al. (2021); Bhat et al. (2021), we decide to present instructions until students have entered a minimum number of words and put a certain amount of delay before showing instructions to minimize potential disruptions caused by irrelevant information from model hallucination (Maynez et al., 2020). Figure 2 presents the system architecture. The student starts with writing the beginning of the review. The system will display instructions until students enter at least 25 words. After this threshold, when the student gets stalled, by pressing the spacebar, they will trigger the model in the backend to generate instructions.

After the keypress, there is a delay of eight seconds before they receive instructions. To preserve the context while avoiding too much overhead for querying the model, we pass the last twenty words from the input to the model. According to UR 4, and supported by Calderwood et al. (2020), overly brief suggestions are often unhelpful. To ensure clarity and concision, we limit each instruction to a maximum of 60 tokens, which is approximately 45 words<sup>8</sup>. In their experiment with one, three, and six instructions, Buschek et al. (2021) discovered that multiple instructions can facilitate the identification of useful phrases and boost their acceptance rate. We decide to present three instructions each time considering the cost-benefit tradeoffs for efficiency (e.g. reading time vs diversified content). The student controls the final output by checking multiple instructions and deciding whether to accept or reject them. They are free to add, delete, and replace the generated content.

<sup>8</sup><https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them>

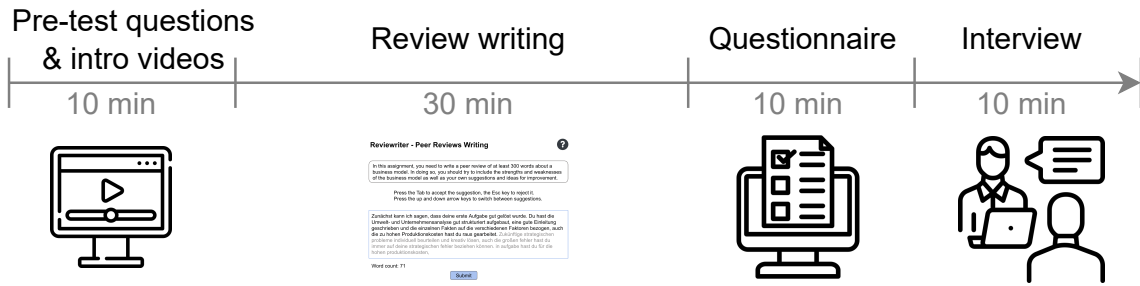


Figure 4: Overview of the study procedure. Students begin with five pre-test questions and two introduction videos. Then, they engage in a 30-minute review writing task. Afterward, they are asked to complete a questionnaire, which is followed by an interview with a set of open-ended questions.

## 4 Evaluation of Reviewriter

### 4.1 Experimental setup

To assess our prototype, we conduct a mixed-method study with fourteen students from a public university in Europe. We reached out to students who have participated in our previous design interview and also recruited students on campus. Fourteen students—eleven males and three females—participated in the evaluation. Three of them were undergraduate students and the rest were graduate students. Four graduate students also participated in our previous design interview. They were all native German speakers and expressed interest in getting AI-generated instructions when writing peer reviews. They have diverse backgrounds, including computer science, robotics, and business with a mean age of 25.33 years ( $SD = 3.60$ ). The evaluation is conducted either face-to-face or remotely with a conference tool. Each student screen records their writing process, the interviews are also recorded and transcribed by a researcher.

1. Pre-test (10 minutes): The experiment starts with a pre-survey that has five questions (Appendix B.1) followed by two videos. The first four questions measure the learners' level of innovation in the field of information technology, following Agarwal and Karahanna (2000). They need to rate their agreement with a statement on a Likert scale ranging from 1 (totally disagree) to 7 (totally agree), with 4 being neutral (Likert, 1932). Following the pre-survey, we present two videos. The first video introduces a business model for a platform that connects ski instructors with learners, and the second video provides guidance on how to use Reviewriter.

2. Peer review writing (30 minutes): In this phase, students are asked to write a review for a peer's business model. Specifically, they are asked to elaborate on strengths, weaknesses, and suggestions for improvement of the given business model. We instruct students not to use search engines and spend a minimum of 15 minutes on the task. A countdown indicates the remaining time.
3. Questionnaire and interview (10+10 minutes): In the post-survey, we ask 29 questions (Appendix B.2) to measure *perceived ease of use*, *perceived ease of interaction*, *perceived level of enjoyment*, *perceived level of excitement* and *perceived usefulness*, following the technology acceptance model of Venkatesh and Bala (2008) and Venkatesh et al. (2003). All constructs are measured with a 1- to 7-point Likert scale. Moreover, we ask several qualitative questions to further examine students' attitudes toward AI-generated instructions and capture the demographics.

### 4.2 Quantitative analysis and qualitative feedback

To measure student perceptions of AI-generated instructions for peer review writing, we calculate the following constructs on a 1- to 7-point Likert scale (Table 3): perceived ease of use ( $M_1 = 6.07$ ,  $SD_1 = 0.83$ ), perceived ease of interaction ( $M_2 = 5.50$ ,  $SD_2 = 1.22$ ), perceived level of excitement ( $M_3 = 5.64$ ,  $SD_3 = 1.15$ ), perceived level of enjoyment ( $M_4 = 5.43$ ,  $SD_4 = 1.16$ ), and perceived usefulness ( $M_5 = 4.64$ ,  $SD_5 = 1.34$ ). The results show that the participants rate positively using Reviewriter to receive adaptive instructions. Moreover, the mean values of the tool are also very promising when comparing the results

Statistics	Perceived ease of use	Perceived ease of interaction	Perceived level of excitement	Perceived level of enjoyment	Perceived usefulness
<b>Mean</b>	6.07	5.50	5.64	5.43	4.64
<b>Std.</b>	0.83	1.22	1.15	1.16	1.34
<b>Normalized mean</b>	0.87	0.79	0.81	0.78	0.66

Table 3: Descriptive statistics from quantitative measure in the evaluation of Reviewriter (N=14). The measure of technology acceptance on a 1 - 7 Likert Scale (1: low, 4: neutral, 7: high).

to the average of the scale. All results are better than the neutral value of four. This fosters motivation and engagement to use the learning application. [Malik et al. \(2021\)](#) found that perceived ease of use ( $M_1 = 6.07$ ) and usefulness ( $M_5 = 4.64$ ) positively influence student adoption intentions and their attitudes toward AI-based applications. The positive levels of perceived ease of interaction ( $M_2 = 5.50$ ), excitement ( $M_3 = 5.64$ ), and enjoyment ( $M_4 = 5.43$ ) suggest that the technology has been accepted favorably. This is especially important for learning tools to ensure students are perceiving the usage of the tool as enjoyable, useful, and easy to interact with ([Marangunić and Granić, 2015](#)). These are promising results for using this tool to receive AI-generated instructions in a peer review setting.

In addition to quantitative scores, we incorporate qualitative open-ended questions to further understand student attitudes toward writing with AI-generated text and how the instructions impact their writing process. We translate the responses from German and cluster the representative ones (Appendix B.3). The general attitude towards Reviewriter was very positive. Five students stated concretely the benefits of Reviewriter on their writing process. Three students mentioned the system is simple and easy to interact with. On the adoption of the generated instructions, one student used them every time, two students stated that they did not find anything useful in the instructions. Another two students reported that they never used the complete instructions but they picked up ideas or keywords from them. Five of them used instructions three to five times, and the rest stated that they use the AI-generated instructions quite frequently and did not provide an exact number. Moreover, it is interesting to note that there are divergent opinions on the delay of the system. Three students complained about the waiting time was too long while two other students were in favor of

the delay and stated that the waiting time left them room to think. Finally, students enjoyed the diverse content in AI-generated instructions while noticing there were ungrammatical sentences and irrelevant phrases from time to time.

## 5 Discussion

Peer review writing is an increasingly important educational task in large-scale or distance learning scenarios since it enables personalized feedback to be delivered at scale, thereby lessening the workload of instructors ([Er et al., 2021](#)) and boosting learners' motivation ([Hsia et al., 2016](#)). However, during writing peer reviews, students may experience obstacles such as writer's block [Rose \(1980\)](#) where they struggle to generate the next line, the right phrase, or the sentence [Oliver \(1982\)](#). LLMs can help to overcome this obstacle by producing adaptive instructions based on students' input, which ultimately aid in the seamless progression of thoughts ([Gero et al., 2022](#)). To do so, we develop a novel peer review writing tool called Reviewriter. It allows students to use AI-generated instructions as an inspiration and incorporate those ideas into their own work in a creative and original way, such as by adapting, mixing, or reinterpreting those instructions ([Qadir, 2022](#)).

Our study contributes at least three key aspects to the innovative use of NLP in education. First, we explore the personalization of AI-generated instructions in a specific pedagogical scenario - peer review writing ([Pardos and Bhandari, 2023](#)) by gathering insights from literature review and student interviews ([Verleger and Pembridge, 2018](#)). Second, in contrast to [Lee et al. \(2022a\)](#) which used GPT-3 without adaptation for collaborative writing, we fine-tune three German language models on a corpus selected based on certain criteria to provide specialized content with high quality. Afterward, we choose German-GPT2 based on quantitative measures and human evaluation. Third, as noted



in [Kuhail et al. \(2023\)](#), "lack of feedback" is one of the challenges to using generative models in education. Therefore, we evaluate our tool with fourteen students and the result reveals positive technology acceptance based on quantitative measures. Through our qualitative evaluation, we find that students generally enjoyed seeing generated instructions with varied content to spark ideas. And they were enthusiastic and excited about writing with generative language models. We recognize that there is a need for further research on the effectiveness of LLM-based writing support tools in various contexts, as well as the improvement of faithfulness and factuality in AI-generated instructions ([Maynez et al., 2020](#)). Nonetheless, our study contributes to the growing body of knowledge on the potential of generative AI to provide personalized writing instructions and enhance students' learning experiences ([Pardos and Bhandari, 2023](#)).

## 6 Conclusion and future work

To help students mitigate writer's block during peer review writing, we design, build, and evaluate *Reviewriter*, a novel tool that aims to provide students with AI-generated instructions during their peer review writing process. We provide design insights with pedagogical considerations of integrating LLMs into peer-review writing systems. Our evaluation involves fourteen students from tertiary education, who reported enjoying the interaction with the system, finding it easy to use, and expressing interest in using similar tools in the future. They also pointed out that the relevance of the generated instructions could be further improved. We present *Reviewriter*, including its design rationales and evaluated interface, as a contribution to the exploration of LLMs' potential in innovative NLP-based approaches in education. As NLP continues to advance, we aspire that our work will encourage other researchers to explore how generative AI can be integrated into educational applications to benefit teachers and students, while promoting responsible and ethical use.

For future work, we will investigate students' perceptions of peer reviews from different sources: their peers, peers using *Reviewriter*, and entirely AI-generated reviews. We will collect ratings and feedback from students who receive these reviews and compare the relevance, quality, and usefulness of the texts generated from each source. Additionally, we aim to integrate *Reviewriter* into

the university's existing peer review system, enabling widespread adoption among students across various courses. By incorporating AI-generated instructions into routine peer reviews, we can examine the long-term impact on student's writing skills, critical thinking abilities, and overall academic performance. To enhance the relevance of the AI-generated instructions in *Reviewriter*, we will refine the algorithms and models based on feedback from our evaluation participants. Our iterative development process will involve incorporating more contextual information, employing advanced NLP techniques, and leveraging user feedback to achieve higher accuracy and helpfulness in the AI-generated instructions.

## References

- Tufan Adiguzel, Mehmet Haldun Kaya, and Fatih Kürşat Cansu. 2023. Revolutionizing education with AI: Exploring the transformative potential of ChatGPT. *Contemporary Educational Technology* 15, 3 (2023), ep429.
- Ritu Agarwal and Elena Karahanna. 2000. Time flies when you're having fun: Cognitive absorption and beliefs about information technology usage. *MIS quarterly* (2000), 665–694.
- Maryam Alqassab, Jan-Willem Strijbos, Ernesto Panadero, Javier Fernández Ruiz, Matthijs Warrens, and Jessica To. 2023. A systematic review of peer assessment design elements. *Educational Psychology Review* 35, 1 (2023), 18.
- Advait Bhat, Saaket Agashe, and Anirudha Joshi. 2021. How do people interact with biased text prediction models while writing?. In *Proceedings of the First Workshop on Bridging Human–Computer Interaction and Natural Language Processing*. Association for Computational Linguistics, Online, 116–121. <https://aclanthology.org/2021.hcinlp-1.18>
- Margaret A Boden et al. 2004. *The creative mind: Myths and mechanisms*. Psychology Press.
- Svend Brinkmann. 2013. *Qualitative interviewing*. Oxford university press.
- Daniel Buschek, Martin Zürn, and Malin Eiband. 2021. The Impact of Multiple Parallel Phrase Suggestions on Email Input and Composition Behaviour of Native and Non-Native English Writers. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) (*CHI '21*). Association for Computing Machinery, New York, NY, USA, Article 732, 13 pages. <https://doi.org/10.1145/3411764.3445372>

- Alex Calderwood, Vivian Qiu, Katy Ilonka Gero, and Lydia B Chilton. 2020. How Novelists Use Generative Language Models: An Exploratory User Study. In *HAI-GEN+ user2agent@ IUI*.
- Zijian Ding Chan et al. 2023. Mapping the Design Space of Interactions in Human-AI Text Co-creation Tasks. *arXiv preprint arXiv:2303.06430* (2023).
- Yu Chen, Scott Jensen, Leslie J Albert, Sambhav Gupta, and Terri Lee. 2023. Artificial intelligence (AI) student assistants in the classroom: Designing chatbots to support student success. *Information Systems Frontiers* 25, 1 (2023), 161–182.
- Elizabeth Clark, Anne Spencer Ross, Chenhao Tan, Yangfeng Ji, and Noah A. Smith. 2018. Creative Writing with a Machine in the Loop: Case Studies on Slogans and Stories. In *23rd International Conference on Intelligent User Interfaces* (Tokyo, Japan) (*IUI '18*). Association for Computing Machinery, New York, NY, USA, 329–340. <https://doi.org/10.1145/3172944.3172983>
- Mike Cohn. 2004. *User stories applied: For agile software development*. Addison-Wesley Professional.
- Harris M Cooper. 1988. Organizing knowledge syntheses: A taxonomy of literature reviews. *Knowledge in society* 1, 1 (1988), 104–126.
- Hai Dang, Karim Benharrak, Florian Lehmann, and Daniel Buschek. 2022. Beyond Text Generation: Supporting Writers with Continuous Automatic Text Summaries. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) (*UIST '22*). Association for Computing Machinery, New York, NY, USA, Article 98, 13 pages. <https://doi.org/10.1145/3526113.3545672>
- Ali Darvishi, Hassan Khosravi, Solmaz Abdi, Shazia Sadiq, and Dragan Gašević. 2022. Incorporating Training, Self-Monitoring and AI-Assistance to Improve Peer Feedback Quality (*L@S '22*). Association for Computing Machinery, New York, NY, USA, 35–47. <https://doi.org/10.1145/3491140.3528265>
- 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, Minneapolis, Minnesota, 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
- Erkan Er, Yannis Dimitriadis, and Dragan Gašević. 2021. Collaborative peer feedback and learning analytics: Theory-oriented design for supporting class-wide interventions. *Assessment & Evaluation in Higher Education* 46, 2 (2021), 169–190.
- Zihan Gao and Jiepu Jiang. 2021. Evaluating Human-AI Hybrid Conversational Systems with Chatbot Message Suggestions. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management* (Virtual Event, Queensland, Australia) (*CIKM '21*). Association for Computing Machinery, New York, NY, USA, 534–544. <https://doi.org/10.1145/3459637.3482340>
- Katy Ilonka Gero, Vivian Liu, and Lydia Chilton. 2022. Sparks: Inspiration for Science Writing Using Language Models. In *Designing Interactive Systems Conference* (Virtual Event, Australia) (*DIS '22*). Association for Computing Machinery, New York, NY, USA, 1002–1019. <https://doi.org/10.1145/3532106.3533533>
- Jochen Gläser and Grit Laudel. 2009. *Expert interviews and qualitative content analysis: as tools for reconstructive research*. Springer-Verlag.
- Lu-Ho Hsia, Iwen Huang, and Gwo-Jen Hwang. 2016. Effects of Different Online Peer-Feedback Approaches on Students' Performance Skills, Motivation and Self-Efficacy in a Dance Course. *Comput. Educ.* 96, C (may 2016), 55–71. <https://doi.org/10.1016/j.compedu.2016.02.004>
- María Soledad Ibarra-Sáiz, Gregorio Rodríguez-Gómez, and David Boud. 2020. Developing student competence through peer assessment: the role of feedback, self-regulation and evaluative judgement. *Higher Education* 80, 1 (2020), 137–156.
- Qinjin Jia, Yupeng Cao, and Edward Gehringer. 2022. Starting from “Zero”: An Incremental Zero-shot Learning Approach for Assessing Peer Feedback Comments. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*. Association for Computational Linguistics, Seattle, Washington, 46–50. <https://doi.org/10.18653/v1/2022.bea-1.8>
- Mohammad Amin Kuhail, Nazik Alturki, Salwa Al-ramlawi, and Kholood Alhejori. 2023. Interacting with educational chatbots: A systematic review. *Education and Information Technologies* 28, 1 (2023), 973–1018.
- Chin-Yuan Lai. 2016. Training nursing students' communication skills with online video peer assessment. *Computers & Education* 97 (2016), 21–30.
- Mina Lee, Percy Liang, and Qian Yang. 2022a. CoAuthor: Designing a Human-AI Collaborative Writing Dataset for Exploring Language Model Capabilities. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) (*CHI '22*). Association for Computing Machinery, New York, NY, USA, Article 388, 19 pages. <https://doi.org/10.1145/3491102.3502030>
- Mina Lee, Megha Srivastava, Amelia Hardy, John Thickstun, Esin Durmus, Ashwin Paranjape, Ines Gerard-Ursin, Xiang Lisa Li, Faisal Ladhak, Frieda Rong,

- et al. 2022b. Evaluating Human-Language Model Interaction. *arXiv preprint arXiv:2212.09746* (2022).
- Hongli Li, Yao Xiong, Xiaojiao Zang, Mindy L. Kornhaber, Youngsun Lyu, Kyung Sun Chung, and Hoi K. Suen. 2016. Peer assessment in the digital age: A meta-analysis comparing peer and teacher ratings. *Assessment & Evaluation in Higher Education* 41, 2 (2016), 245–264.
- Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of psychology* (1932).
- Hui-Chen Lin, Gwo-Jen Hwang, Shao-Chen Chang, and Yaw-Don Hsu. 2021. Facilitating critical thinking in decision making-based professional training: An online interactive peer-review approach in a flipped learning context. *Computers & Education* 173 (2021), 104266.
- Reena Malik, Ambuj Shrama, Sonal Trivedi, and Rik-kee Mishra. 2021. Adoption of Chatbots for learning among university students: role of perceived convenience and enhanced performance. *International Journal of Emerging Technologies in Learning (iJET)* 16, 18 (2021), 200–212.
- Nikola Marangunić and Andrina Granić. 2015. Technology acceptance model: a literature review from 1986 to 2013. *Universal access in the information society* 14, 1 (2015), 81–95.
- 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*. Association for Computational Linguistics, Online, 1906–1919. <https://doi.org/10.18653/v1/2020.acl-main.173>
- Piotr Mirowski, Kory W Mathewson, Jaylen Pittman, and Richard Evans. 2022. Co-writing screenplays and theatre scripts with language models: An evaluation by industry professionals. *arXiv preprint arXiv:2209.14958* (2022).
- Lawrence J Oliver. 1982. Helping students overcome writer’s block. *Journal of Reading* 26, 2 (1982), 162–168.
- Zachary A Pardos and Shreya Bhandari. 2023. Learning gain differences between ChatGPT and human tutor generated algebra hints. *arXiv preprint arXiv:2302.06871* (2023).
- Francesc Pedróf, Miguel Subosa, Axel Rivas, and Paula Valverde. 2019. Artificial intelligence in education : challenges and opportunities for sustainable development.
- Junaid Qadir. 2022. Engineering education in the era of ChatGPT: Promise and pitfalls of generative AI for education. (2022).
- 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 (2019), 9.
- Roman Rietsche, Andrew Caines, Cornelius Schramm, Dominik Pfütze, and Paula Buttery. 2022. The Specificity and Helpfulness of Peer-to-Peer Feedback in Higher Education. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*. Association for Computational Linguistics, Seattle, Washington, 107–117. <https://doi.org/10.18653/v1/2022.bea-1.15>
- Roman Rietsche, Kevin Duss, Jan Martin Persch, and Matthias Soellner. 2018. Design and Evaluation of an IT-based Formative Feedback Tool to Foster Student Performance. In *Proceedings of the International Conference on Information Systems (ICIS)*. San Francisco, CA, USA.
- Roman Rietsche and Matthias Söllner. 2019. Insights into Using IT-Based Peer Feedback to Practice the Students Providing Feedback Skill. Proceedings of the Hawaii International Conference on System Sciences (HICSS), Maui, HI, USA.
- Mike Rose. 1980. Rigid Rules, Inflexible Plans, and the Stifling of Language: A Cognitivist Analysis of Writer’s Block. *College Composition and Communication* 31, 4 (1980), 389–401. <http://www.jstor.org/stable/356589>
- Philip M Sadler and Eddie Good. 2006. The impact of self-and peer-grading on student learning. *Educational assessment* 11, 1 (2006), 1–31.
- Teven Le Scao, Angela Fan, Christopher Akiki, El-lie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. *arXiv preprint arXiv:2211.05100* (2022).
- Hua Shen and Tongshuang Wu. 2023. Parachute: Evaluating Interactive Human-LM Co-writing Systems. *arXiv preprint arXiv:2303.06333* (2023).
- Eva AM van Dis, Johan Bollen, Willem Zuidema, Robert van Rooij, and Claudi L Bockting. 2023. ChatGPT: five priorities for research. *Nature* 614, 7947 (2023), 224–226.
- Viswanath Venkatesh and Hillol Bala. 2008. Technology acceptance model 3 and a research agenda on interventions. *Decision sciences* 39, 2 (2008), 273–315.
- Viswanath Venkatesh, Michael G Morris, Gordon B Davis, and Fred D Davis. 2003. User acceptance of information technology: Toward a unified view. *MIS quarterly* (2003), 425–478.

- Matthew Verleger and James Pembroke. 2018. A pilot study integrating an AI-driven chatbot in an introductory programming course. In *2018 IEEE frontiers in education conference (FIE)*. IEEE, 1–4.
- Thiemo Wambsganss, Andrew Caines, and Paula Buttery. 2022a. ALEN App: Argumentative Writing Support To Foster English Language Learning. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*. Association for Computational Linguistics, Seattle, Washington, 134–140. <https://doi.org/10.18653/v1/2022.bea-1.18>
- Thiemo Wambsganss, Christina Niklaus, Matthias Cetto, Matthias Söllner, Siegfried Handschuh, and Jan Marco Leimeister. 2020. AL: An Adaptive Learning Support System for Argumentation Skills. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (*CHI '20*). Association for Computing Machinery, New York, NY, USA, 1–14. <https://doi.org/10.1145/3313831.3376732>
- Thiemo Wambsganss, Christina Niklaus, Matthias Söllner, Siegfried Handschuh, and Jan Marco Leimeister. 2021. Supporting cognitive and emotional empathic writing of students. *arXiv preprint arXiv:2105.14815* (2021).
- Thiemo Wambsganss, Vinitra Swamy, Roman Rietsche, and Tanja Käser. 2022b. Bias at a Second Glance: A Deep Dive into Bias for German Educational Peer-Review Data Modeling. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na (Eds.). International Committee on Computational Linguistics, 1344–1356. <https://aclanthology.org/2022.coling-1.115>
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, Brussels, Belgium, 353–355. <https://doi.org/10.18653/v1/W18-5446>
- 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*. Association for Computational Linguistics, Online, 38–45. <https://doi.org/10.18653/v1/2020.emnlp-demos.6>
- Wei Xu, Marvin J Dainoff, Liezhong Ge, and Zaifeng Gao. 2021. From human-computer interaction to human-AI interaction: new challenges and opportunities for enabling human-centered AI. *arXiv preprint arXiv:2105.05424* 5 (2021).
- Daijin Yang, Yanpeng Zhou, Zhiyuan Zhang, Toby Jia-Jun Li, and Ray LC. 2022. AI as an Active Writer: Interaction strategies with generated text in human-AI collaborative fiction writing. In *Joint Proceedings of the ACM IUI Workshops*, Vol. 10.
- Olaf Zawacki-Richter, Victoria Marín, Melissa Bond, and Franziska Gouverneur. 2019. Systematic review of research on artificial intelligence applications in higher education -where are the educators? *International Journal of Educational Technology in Higher Education* 16 (10 2019), 1–27. <https://doi.org/10.1186/s41239-019-0171-0>

## A Details on data pre-processing and models

### A.1 Template questions asked students to write reviews which some students copied to their review text

- What do you see as the strengths of the fellow student’s solution?
- What do you see as weaknesses in the fellow student’s solution and how can they be addressed?
- What should be paid attention to in the revision of the solution?
- Provide concrete suggestions for improvement in this regard.
- Give concrete suggestions for improvement (constructive feedback).
- What should you pay attention to in the revision of the solution? Give concrete suggestions for improvement (constructive feedback).

### A.2 Abbreviations and expansions

Abbreviation	Expansion
bsp, bspw	beispielsweise
dh	da her
ev, evtl	eventuell
ggf	gegebenenfalls
oä	oder ähnliches
vlt	vielleicht
zb	zum Beispiel

Table 4: A list of abbreviations students used in the review text and we replace with the expansion in the pre-processing.

### A.3 Hyperparameters for pretrained language models

Hyperparameter	GPT2	BLOOM
Vocabulary size	50257	250880
Attention heads	12	8
Hidden layers	12	2
Attention dropout	0.1	0.1

Table 5: Hyperparameters for pretrained GPT2 and BLOOM

## A.4 Sample text generated by different language models

### B Details on evaluations

#### B.1 Pre-test questions asked during evaluation of Reviewriter

1. I like experimenting and trying out new technologies.
2. As a rule, I am hesitant when trying out new technologies.
3. In my circle of friends, I’m usually the first person to try new digital media / new technologies.
4. When I hear about new technologies I look for a way to experiment with them.
5. I have had experience writing reviews/feedback in the past.

#### B.2 Post-test questions asked during evaluation of Reviewriter

- Transition questions: How many times have you accepted Reviewriter’s recommendations?
- Technology Acceptance Model
  1. Assuming the review writing assistance tool is available, the next time I want to write a review/feedback I would use it again.
  2. With Reviewriter I can write reviews/feedback more effectively.
  3. Learning to use Reviewriter was easy for me.
  4. I find using Reviewriter useful for writing reviews/feedbacks.
  5. I find Reviewriter easy to interact with.
  6. It would be easy for me to become familiar with Reviewriter.
  7. Compared to other participants, I think I wrote a very convincing review/feedback.
  8. After using Reviewriter, my ability to write reviews/feedback has improved.
  9. I’m sure I wrote a very insightful review/feedback.
  10. I’m sure I wrote a very convincing review/feedback.

11. With Reviewriter I can write better reviews/ feedbacks.
  12. I think I now know more about how to write well-structured, persuasive, and insightful reviews/feedbacks.
  13. Assuming Reviewriter was available, the next time I write a review/feedback I would use it.
  14. After using Reviewriter, my ability to pay attention to the different parts of the review/feedback structure has improved.
- Evaluate student perceptions on the AI-generated instructions
    1. I expect Reviewriter will help me improve my ability to write well-structured reviews/feedbacks.
    2. I assume Reviewriter would help me improve my ability to write compelling reviews/feedback.
    3. I assume Reviewriter would help me improve my ability to write insightful reviews/feedback.
    4. Interacting with the tool was fun and enjoyable for me.
    5. I expect Reviewriter will help me improve my ability to write helpful reviews/feedback.
    6. Interacting with the tool was exciting.
  - Open-ended questions for qualitative feedback
    1. How has Reviewriter impacted your writing process?
    2. What did you particularly like about using Reviewriter?
    3. Do you have any other ideas?
    4. What could still be improved?
    5. Have you used a writing support program before (e.g. Grammarly)?
    6. What is your field of study?
    7. Please enter your gender.
    8. Please indicate your mother tongue.

**B.3 Clustered qualitative student feedback from the evaluation of Reviewriter**

Source	Text
Original input	First of all, I can say that your first task was solved well. You have structured the environmental and company analysis well, written a good introduction, and related the individual facts to the various factors, and you have also worked out the excessively high production costs.
German GPT-2 generated instructions	Furthermore, the problems were identified and their elaboration was well coordinated. However, the behaviour of this company is completely unnecessary and one could hardly judge it more. Unfortunately, you went into far too much detail about the individual points and did little about the environmental and company analysis. 2) What do you see as the weaknesses of the environmental and business analysis and how can these be remedied?1.written out above:However, with the exception of case-related, these are other generic terms that have a rather different linguistic form.
BLOOM generated instructions	I have already recognized the second task and the reasoning is also understandable for you. Good luck for the future! Strengths) The project idea is cleanly and realistically implemented, as it is a
GerPT2-large generated instructions	You mastered the task very well. I have a few suggestions for improvement. First of all, you could have explained in the assignment what an environmental and company analysis actually is. I would also replace the word "company" with "firm" or something similar. You answered task two well, but you might have briefly explained the stakeholder approach, which was necessary to understand the task. Perhaps you could have changed the focus of the company analysis, but that is only a small point. In general you could

Table 6: Comparison of the sample text generated by different transformer-based pretrained language models with max\_length=150 (we translated all text from German to English for the purpose of this paper).

Topic	Cluster	Statement
On the adoption of the AI-generated instructions	Positive	S1: "I mainly accepted the ideas and slightly rewrote the proposed text." S3: "I find myself be inspired by professional keywords." S11: "I used the recommendations every time."
	Constructive	S4: "Never. They were utterly useless."
On the quality of the AI-generated instructions	Positive	S1: "A few of the suggested ideas were very relevant. It also often remind me to say something positive." S4: " I like that it suggests diverse ideas that are quite different from each other." S10: "Reviewriter provided me with novel ideas that I could explore in depth."
	Constructive	S1: "Shorter instructions would be more relevant sometimes." S10: "The instructions sometimes have spelling mistakes." S11: " Sometimes I got instructions that didn't fit the content." S12: "I would suggest to generate shorter snippets. Sometimes the beginning wasn't bad but later it got weird."
On the impact of the writing process	Positive	S2: "The tool helps break through writer's block." S3: " When I got stuck on what to write, it sometimes had useful keywords, which made me a little quicker." S10: "The review writing process has accelerated." S11: "I got new ideas from Reviewriter's suggestions. I think the system not only helps to write structured reviews, but also to come up with new ideas. This is where I see the greatest potential." S14: " I didn't feel so alone while writing."
	Constructive	S1: "Waiting for suggestions slowed down my writing process." S12: "I tried to adopt the instructions a couple of times to be more efficient. However, since the waiting time for the instructions is very long, the process has been delayed."
On the system interaction	Positive	S5, S8: "It is easy to use and simple to operate." S10: "It is easy to use and saves time." S11: "I liked that I was not forced to accept the instructions and I could choose among several options."
	Constructive	S11: "I think it would be better if we could select the instructions with the mouse."
On the delay of instructions	Positive	S2: "Latency is moderate." S9: "I did not get suggestions instantaneously, I really just got it when I wanted it. That was really good, because that way my thoughts did not get interrupted." S14: "It is good that the instructions don't come immediately after I stop writing. It didn't disrupt my flow of writing."
	Constructive	S6: "The proposals come too late, I almost come up with my own ideas." S1, S10, S12: "The waiting time for suggestions is long."

Table 7: We have categorized the qualitative feedback received from fourteen students (referred to as S1 to S14) from tertiary education, who participated in the evaluation of Reviewriter. We collected the feedback through open-ended questions in the post-survey and concluding interview. For qualitative questions answered in German, we translated the written responses into English. The interview was conducted in English, recorded with the students' consent.

# Towards L2-friendly pipelines for learner corpora: A case of written production by L2-Korean learners

**Hakyung Sung**

Department of Linguistics  
University of Oregon  
hsung@uoregon.edu

**Gyu-Ho Shin**

Department of Linguistics  
University of Illinois Chicago  
gyuhoshin@gmail.com

## Abstract

We introduce the Korean-Learner-Morpheme (KLM) corpus, a manually annotated dataset consisting of 129,784 morphemes from second language (L2) learners of Korean, featuring morpheme tokenization and part-of-speech (POS) tagging. We evaluate the performance of four Korean morphological analyzers in tokenization and POS tagging on the L2-Korean corpus. Results highlight the analyzers' reduced performance on L2 data, indicating the limitation of advanced deep-learning models when dealing with L2-Korean corpora. We further show that fine-tuning one of the models with the KLM corpus improves its accuracy of tokenization and POS tagging on L2-Korean dataset.

## 1 Introduction

The use of learner corpora has played a crucial role in understanding language learners' developmental aspects (e.g., Biber et al., 2011; Ellis and Ferreira-Junior, 2009; Gablasova et al., 2017). With the recent advancement of computational methods and techniques, automatic processing of learner corpora (together with sizeable datasets) is gaining momentum for a better understanding of the properties of learner language (e.g., Bestgen and Granger, 2014; Kyle and Crossley, 2017; Lu, 2010).

Despite the increasing interest in this approach, we identify two major caveats in the current research practice. One is the sampling bias towards dominant/hegemonic viewpoints and discourse, especially centering around a limited range of languages and language-usage contexts (e.g., L2 English) (c.f., Bender et al., 2021). This poses a threat to linguistic diversity, equity, and inclusion in the field, as well as weakening the generalizability of previous findings to other (and lesser-studied) languages.

The other caveat concerns the degree to which first language (L1)-based automatic processing pipelines work for L2 data. Indeed, a line of research has questioned the reliability of currently existing parsing/tagging models, which are trained and tested exclusively on the basis of L1 data, when applied to L2 corpora (e.g., Kyle, 2021; Meurers and Dickinson, 2017). This is because these L1-oriented models may not fully account for the characteristics of learner language, including spacing/spelling errors and novel combinations of words and phrases. These factors may negatively impact the performance of L1-based tools when analyzing linguistic features of L2 corpora, thus necessitating empirical investigation.

In this study, we aim to address these caveats by developing a sizable L2-Korean corpus, featuring enhanced morpheme tokenization and POS tagging of the open-access L2-Korean corpus dataset, which comprises 129,784 morphemes (7,527 sentences). Using this dataset, we evaluate the morpheme tokenization and POS-tagging accuracy of two language-general parsers incorporating cutting-edge algorithms (*Stanza*, *Trankit*) and two Korean-specific parsers commonly used by researchers in Korean studies (*Kkma*, *Komorán*).

This paper is structured as follows: We discuss the significance of morphological analysis in Korean studies and review relevant L2-Korean applied research. Next, we outline the annotation process employed in our study. We then elaborate on our methodology for evaluating the performance of the morpheme analyzers on our dataset, using an L1 corpus as a reference. Following this, we present a comprehensive analysis of the overall performance, including detailed comparisons across different proficiency levels and POS tags, as well as a re-evaluation of performance after training the L2 annotated corpus. Finally, we summarize our findings and propose future directions.



## 2 Background

### 2.1 Linguistic properties of Korean

Korean, a language typologically distinctive from the major languages studied in the field (specifically English) is characterized by its agglutinative nature and Subject Object Verb word order. It features overt case-marking and active suffixation, allowing scrambling and omission of sentential components contingent upon contexts (Sohn, 1999). These characteristics collectively pose challenges to automatic processing of (L2-)Korean corpora (Shin and Jung, 2021), particularly for tokenization and POS tagging systems that are not entirely rely on white-space units such as English words (McDonald et al., 2013). Previous studies (e.g., Choi and Palmer, 2011; Park et al., 2013) have addressed word-level representation issues in Korean by utilizing linguistically motivated rules, highlighting the fact that words in Korean comprise both lexical and functional *morphemes* (i.e., the smallest meaningful unit of language). This necessitates considering morpheme-level parsing and tagging when handling Korean corpora automatically (Chen et al., 2022).

### 2.2 Application of morpheme tokenizers and POS taggers in L2-Korean research

In spite of the language-specific challenges associated with Korean for conducting automatic text processing, researchers have increasingly attempted to apply NLP techniques to L2-Korean research. Notably, however, most studies have not provided sufficient information about the tools they used or the reliability of the parsers/taggers for L2-text processing. An overview of this research practice is outlined below.

**Error analysis:** Kim et al. (2016) investigated the types of frequent errors from a sizable L2-Korean writing data ( $n=500$ ) and identified rules for searching syntactic patterns by using a POS tagger (type not reported). Lee et al. (2016) proposed an automatic error-detection scheme for L2-Korean production involving functional morphemes (e.g., particles) in combination with a POS tagger (type not reported).

**Lexico-grammatical token measurement:** Lim et al. (2022) proposed an automated writing evaluation system by employing a transformer-based multilingual model and XLM-RoBERTa.

They used a POS tagger (type not reported) to measure the number of morphemes as one of the complexity features of learner writing. Nam and Hong (2014) collected L2-Korean spoken data from storytelling, communications, and natural conversations and annotated the data based on the Sejong tag set. They employed a POS tagger (type not reported) to compare the number of particles across multiple proficiency groups.

**Morpheme/construction extraction:** Jung (2022) and Shin and Jung (2022) investigated the distribution of Korean particles in L2-Korean textbooks. Using *UDpipe* as a tagger, they developed a pipeline for automatically extracting the target particles. Likewise, Shin and Jung (2021) demonstrated how Korean passive constructions could be (semi-)automatically identified by using the same tagger and pipeline developed above.

**Text similarity analysis:** Cho and Park (2018) used various morphological analyzers (*Kkma*, *Okt*, *Hannanum*, and *Komorán*) to explore the text similarity (based on TF-IDF) of the writings produced by sixteen different L2-Korean learners.

## 3 Dataset

The Korean-Learner-Morpheme (KLM) corpus, as it currently stands, comprises 129,784 morphemes (67,284 *eojeols*, which are sequences of Korean characters separated by white-spaces) with morpheme tags grounded in the Sejong tag set (Appendix A). This corpus was sourced from the Kyung Hee Korean learner written corpus collected by Park and Lee (2016). The corpus encompasses data on classroom proficiency levels (ranging from 1 to 6 as a proxy for learner proficiency), nationality, gender, and writing topics. To create our dataset, we randomly extracted a total of 600 texts from the original corpus, with each proficiency level represented by 100 texts.

Despite the presence of morpheme tokenization and POS tags in the original corpus, several issues prevented its direct use for evaluation purposes, which ultimately led us to conduct manual annotations. First, without gold annotations for the data, we were not able to determine the accuracy of the automatic POS tagger (i.e., ESPRESSO) that Park and Lee (2016) used for morphological analysis. Additionally, we were uncertain whether the annotation scheme in the original cor-

pus had been thoroughly tested, taking into account the language-specific properties of Korean. Second, we were unsure how the characteristics of learner language (e.g., spelling/spacing errors), which were not clearly indicated in the original corpus, were documented in the annotations (e.g., whether they were corrected or neglected during the automatic analysis). On top of these issues, the formatting proved difficult to process the data automatically.

To create our corpus, we first reformatted the texts into CoNLL-U format, following the Universal Dependencies (UD) formalism (c.f., Nivre et al., 2020). To ensure the metadata in the original dataset, we associated the respective # text\_id attribute with the extracted metadata (e.g., # text\_id = A100000\_v01\_중국\_남자\_사진기\_빌리기) and incorporated the # sent\_id attribute in an incremental manner (e.g., # sent\_id = A100000\_v01\_중국\_남자\_사진기\_빌리기\_1, assigned to the first sentence of the text) for data management. Sentence- and eojeol- level segmentations were done using Stanza<sup>1</sup> as a tokenizer.

### 3.1 Annotation procedure

The corpus was annotated by two native Korean speakers: the first author of the paper and a graduate student who majored in Korean during their undergraduate studies. Before annotating the sentences, both annotators familiarized themselves with the Sejong tag set, its tokenization scheme<sup>2</sup>, and the annotation guidelines from previous studies related to Korean UD guidelines (e.g., Chun et al., 2018; Park and Tyers, 2019) through two training sessions. The annotation process was carried out in the following steps: (1) the two annotators annotated 100 texts individually (both morpheme tokenization and POS tagging); (2) the annotators reviewed and discussed their disagreements; (3) if a disagreement was not resolved, the third annotator, the second author of this paper, reviewed the problematic tokens and POS tags and provided annotations; and (4) the third annotator commented on the entire annotation results, which were then discussed by the two main annotators before starting the next annotation round.

Although the annotators referred to previous studies for parsing/tagging guidance, there were a few instances in which making deci-

sions proved challenging. Below are the major cases that we discussed, with the purpose of consistent annotations and better evaluation of morpheme tokenizers/taggers of interest. The full tagging guidelines and examples can be accessed here for related future projects: [https://github.com/NLPxL2Korean/Korean\\_Learner\\_Morpheme\\_corpus](https://github.com/NLPxL2Korean/Korean_Learner_Morpheme_corpus).

**Causative and passive markers:** Causative and passive voices are often indicated by the voice markers (-i/hi/li/ki/wu/kwu/chwu- for morphological causative; -i/hi/li/ki- for suffixal passive; -e/a ci- for periphrastic passive; Sohn, 1999). These morphemes, when attached to a root, form causative or passive verbs and lead to changes in valence (i.e., the number of arguments controlled by a predicate in a clausal construction). We parsed all relevant morphemes and assigned them XSV (Suffix, verb derivative) POS tags (e.g., *mek+ta* "to eat" VV (Verb, main)+EF (Ending, closing); *mek+hi+ta* "to be eaten" VV+XSV+EF).

**Auxiliary verbs:** Verbs such as *iss-* "to be/exist/have", *ha-* "to do", and *toy-* "to become" function as both main verbs and auxiliary verbs. As main verbs, they typically operate independently, representing concepts of existence, activity, or possession (e.g., *ku-nun cha-ka iss-ta* "He has a car"). In these instances, we assigned a VV (Verb, main) tag. Conversely, when serving as auxiliary verbs, they work in conjunction with a main verb to convey grammatical meanings, such as continuous or progressive actions (e.g., *ku-nye-nun chayk-ul ilk-ko iss-ta* "She is reading a book"). In these cases, we assigned a VX (Verb, auxiliary) tag.

**Copula, positive:** The copula (-i) is a grammatical element that links the subject of a sentence with a predicate, often conveying a positive meaning (VCP). One complexity in parsing morphemes arises when the copula is combined with the ending *-lanun* in a compound form. This combination links the subject of a sentence to a noun or descriptive phrase while adding a nuance of specification, identification, or definition (translated as "called," "named," or "known as" in English). Interestingly, in some cases, the copula may be hidden, requiring the addition of *-i* before the ending *-lanun* to ensure accurate parsing (e.g.,

<sup>1</sup><https://github.com/stanfordnlp/stanza/>

<sup>2</sup>publicly available from KoNLPy website <https://konlpy.org/ko/v0.4.4/morph/>

*swukcey-lanun* "(the thing) called homework" → *swukcey+i+lanun*, NNG+VCP+ETM).

**Spelling errors:** Instead of judging or omitting the annotation of misspelled words based on annotators’ subjective interpretations, we opted for assigning three relevant tags from the Sejong tag set: NA (Undefined), NF (Undefined, but considered a noun), and NV (Undefined, but considered a verb). Following this annotation method, a total of 2,289 errors were marked (NA: 738, NF: 1,290, NV: 261).

### 3.2 Annotation review

Table 1 presents (1) the number and percentage of refined tokens and tags, and (2) the number and percentage of overall agreement rates between the two annotators in creating the corpus. The term "refined" tokens and tags refers to tokens and tags which were manually revised by the annotators against the tokens and tags used in the original corpus. Note that morpheme tokenization/POS tagging is not always a binary decision in Korean, as the morpheme boundary can be ambiguous. Therefore, we measured the reliability by calculating the ratio of the number of agreement items to the total number of tokens/tags, rather than by calculating Cohen’s Kappa scores. Overall, the results indicate a high level of agreement between the annotators in both tasks.

Category	Token	Tags
# of refinement	19,481	20,987
% of refinement	15.01	16.17
# of agreement	128,890	128,243
% of agreement	99.31	98.81
<b>Total</b>		129,784

Table 1: Summary of annotation results

## 4 Analysis

### 4.1 Reference L1 corpus

We used the Google Korean Universal Dependency Treebank (UD Korean GSD) as a reference L1 corpus to establish a baseline for calculating accuracy. This dataset originally comprises around 6,000 sentences sourced from online blogs and news produced by Korean native speakers. The sentences were then annotated according to the UD guidelines (McDonald et al., 2013) and later enhanced by implementing a more refined morpheme tok-

enizations (Chun et al., 2018). For the purposes of this study, we employed 989 sentences from the UD Korean GSD test set.

### 4.2 Morphological analyzers

We employed four open-access morphological analyzers. They are based on various computational algorithms, ranging from statistical models<sup>3</sup>, which have been widely used by L2-Korean researchers (e.g., Kkma<sup>4</sup>, Komoran), to deep-learning models such as Stanza<sup>1</sup> and Trankit<sup>5</sup>.

## 5 Results and Discussion

### 5.1 Overall performance

Table 2 displays the overall F1 scores<sup>6</sup> of the morphological analyzers for the L2 (target) and L1 (reference) datasets<sup>7</sup>. Figure 1 presents by-proficiency-level performance per analyzer.

Analyzer	Token		Tag	
	L2	L1	L2	L1
Stanza	<b>0.89</b>	0.92	<b>0.86</b>	0.93
Trankit	0.81	0.85	0.80	0.88
Kkma	0.86	0.88	0.80	0.81
Komoran	<b>0.89</b>	0.92	<b>0.86</b>	0.86

Table 2: F1 scores (overall)

We draw three main observations. First, all the analyzers exhibited reduced performance on the

<sup>3</sup>KoNLPy as an interface, see Park and Cho, 2014

<sup>4</sup>Kkma employs a more extensive tag set (52 tags) compared to the other three analyzers (45 tags from the Sejong tag set), necessitating an additional step for tag standardization prior to evaluating accuracy.

<sup>5</sup><https://github.com/nlp-uoregon/trankit/>

<sup>6</sup>It is often the case that True Negatives apply to a binary classification problem in which tokenization is clearly based on white-space, such as English. Notably, tokenization in Korean does not always fall into binary classification because of unclear morpheme boundaries. We thus calculated the F1 scores using True Positives (the number of correct matches between the predicted and gold standard annotations), False Positives (the number of predicted annotations that do not match the gold standard annotations), and False Negatives (the number of gold standard annotations that do not match the predicted annotations). We acknowledge that our approach here should be further verified by future research with providing a *Perfect* matrix (Raman et al., 2022).

<sup>7</sup>Subtle differences in output representation arise when comparing the performance of Stanza/Trankit to that of Kkma/Komoran. Stanza/Trankit utilize word-level units based on white-space, facilitating a robust comparison between annotated and predicted tags, as their outputs are structured around these word-level units. On the other hand, Kkma/Komoran display morphemes without maintaining original word boundaries, necessitating the evaluation of accuracy strictly on a sentence-unit level.

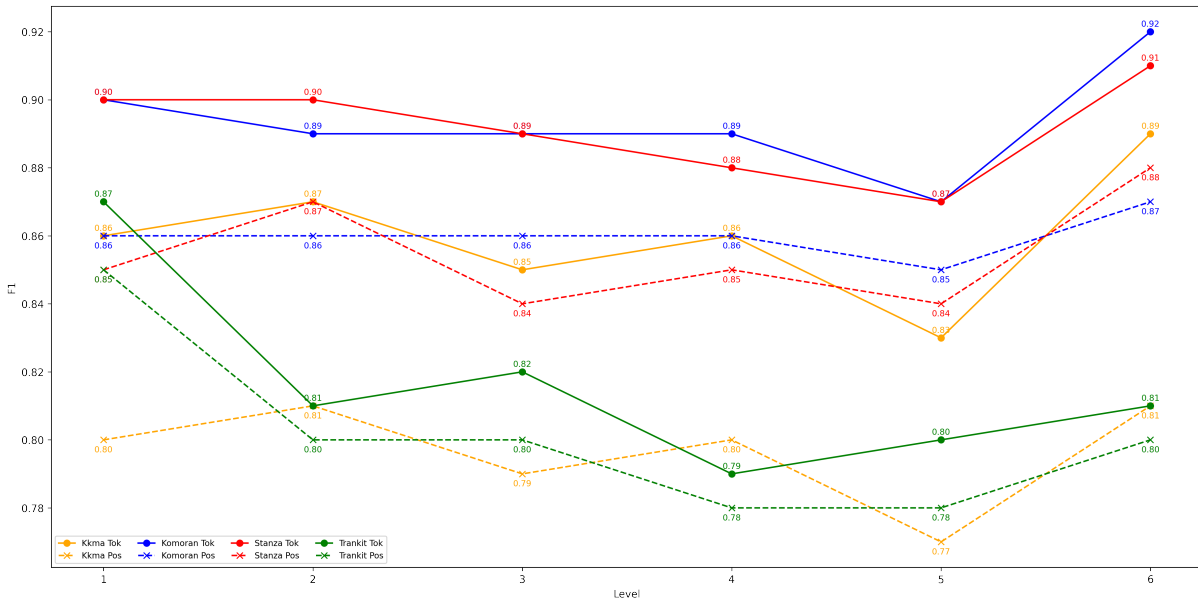


Figure 1: Comparison of analyzers (by-level) in L2 dataset

L2 data compared to their performance on the L1 data, indicating the challenges to automatic L2-data processing induced by learner language characteristics (and possibly in conjunction with the linguistic properties of Korean). Second, Stanza and Komoran achieved the highest F1 scores (tied) in morpheme tokenization and POS tagging on the L2 data. Given that Stanza and Trankit utilize state-of-art deep-learning algorithms, while Komoran is based on a comparatively basic probabilistic model, this finding indicates that even sophisticated models may suffer from coping with Korean learner corpora. Third, each analyzer demonstrated asymmetric patterns of performance by proficiency level. To illustrate, whereas the accuracy rates of Stanza and Komoran remained relatively stable across the levels, the accuracy rate of Trankit decreased notably after Level 2 (novice-intermediate). Of the four analyzers, Kkma showed the largest gap between the tokenization accuracy and the POS-tagging accuracy for all the levels.

## 5.2 By-tag performance

To examine the variation in performance across individual tags within the given datasets, we conducted a comparative analysis between the best-performing models (Stanza, Komoran) for each tag, as shown in the second and third columns of Table 3 (only includes results for the L2 data; see Appendix B for information on the L1 data). To calculate the by-tag accuracy, we included only the cases in which the number of predicted tags and the

number of annotated tags were the same (within an eojeol unit for Stanza; within a sentence unit for Komoran). This approach ensures a fair comparison by maintaining an equal number of tags, avoiding any mismatch that could affect the evaluation process in an unexpected/uncontrollable way. Consequently, there was a discrepancy between the two tokenizers in terms of the number of tags ultimately included in the analysis.

To keep our analysis concise, we excluded tags related to punctuation, numbers, foreign languages, and errors, as well as tags with a low frequency count (overall counts below 10), resulting in a total of 29 tags for the main analysis. In the following section, we discuss tags with low accuracy or those that were of particular interest in previous studies. We also present the confusion matrix for these tags calculated by Stanza in Figure 2a, in which the off-diagonal elements indicate the number of incorrect predictions.

**Predicate-related tags:** The accuracy of VV (Verb, main), VX (Verb, auxiliary), and VA (Verb, adjective) was not satisfactory (except for VA in Komoran). This finding is surprising when we consider the status of verb and adjective as the primitive syntactic categories in human language and as one of the most significant content morphemes in Korean. Upon examining the confusion matrix (Figure 2a), we observed a considerable number of mismatches among these three groups,

with a majority of the VX tags being predicted as the VV tags. The verb *iss-* emerged as one that requires further refinement in future research regarding its POS tags, because its classification as either a VV or VX, depending on its formal co-occurrences with other morphemes, was not effective. Overall, these results suggest that the distinctions between main verbs, adjectives, and auxiliary verbs may not be clear-cut with the current taggers. These ambiguities could stem from linguistic complexities, overlapping grammatical features, or limitations in the underlying model’s ability to discern the subtle differences between them.

**Noun-related tags:** XR (Noun, root) and NP (Pronoun) demonstrated notable by-analyzer asymmetries. Caution is needed, however, as their occurrences in the dataset were small. Considering language-specific properties of Korean (e.g. pronoun are underused), further investigation is required with a more sizeable dataset to fully reveal model performance on these tags.

**Particle- and suffix-related tags:** Particles and suffixes are often considered challenging for the automatic processing of Korean (Shin and Jung, 2021). The results demonstrate that most particle-related tags (JKO, JKS, JKG, JKB, JX; but except for JC) and some suffix-related tags (predicate ending: EF, EC, EP) exhibited relatively high accuracy (mostly above 0.85) whereas tags comprising X (derivational suffixes: XSA, XSN, XSV) seemed not. The confusion matrix revealed that XSA was often tagged as XSV, and XSV as EC.

### 5.3 Model training through L2 data

Based on these observations, we trained a model on an L2 dataset and evaluated if model performance improved in comparison to a model trained solely on an L1 dataset. To construct the model, we split the KLM corpus into three datasets (80% for a training set; 10% for a development/validation set; 10% for a test set) and employed Stanza (pre-trained on the UD Korean GSD training set) to train morpheme tokenization (i.e., lemma) and tagging (i.e., XPOS) annotation models. For training the POS/morphological features tagger modules, we employed pre-trained embedding vectors from the L1-Korean-GSD model and integrated our L2 test

dataset to the vector space. The accuracy evaluation was performed using the L1/L2 test sets with gold standard tokenization and POS tagging.

Analyzer	Stanza (count)	Komorán (count)	Stanza+L2 (count)
JKO	0.94 (4705)	0.93 (2212)	<b>0.96</b> (454)
MAJ	<b>0.94</b> (1192)	<b>0.94</b> (668)	0.85 (143)
JKS	0.92 (4160)	0.91 (1874)	<b>0.95</b> (402)
JKG	0.92 (1257)	0.85 (423)	<b>0.95</b> (119)
EF	0.91 (7389)	<b>0.99</b> (3583)	0.93 (730)
VCN	0.91 (178)	<b>0.95</b> (75)	0.86 (26)
JKB	0.89 (6399)	0.89 (423)	<b>0.92</b> (634)
EC	0.88 (8871)	<b>0.90</b> (3920)	<b>0.90</b> (846)
MAG	0.87 (4628)	<b>0.90</b> (1885)	0.86 (446)
ETM	0.86 (6843)	0.90 (2753)	<b>0.91</b> (689)
JX	0.86 (5317)	<b>0.91</b> (2384)	<b>0.91</b> (543)
EP	0.86 (2984)	<b>0.98</b> (1299)	0.87 (289)
NNB	<b>0.85</b> (4685)	0.84 (1887)	0.84 (532)
XSN	0.84 (1557)	0.85 (581)	<b>0.87</b> (139)
ETN	0.83 (831)	<b>0.89</b> (326)	0.85 (83)
NNG	0.77 (30353)	0.82 (9682)	<b>0.83</b> (2866)
VCP	0.80 (2307)	<b>0.89</b> (744)	0.85 (216)
VV	0.74 (12704)	0.82 (4672)	<b>0.85</b> (1073)
MM	0.76 (1799)	<b>0.89</b> (733)	0.81 (223)
JC	0.77 (712)	0.63 (287)	<b>0.80</b> (61)
XSV	0.75 (3956)	<b>0.85</b> (1705)	<b>0.85</b> (364)
VA	0.73 (4028)	<b>0.92</b> (1547)	0.81 (392)
NP	0.68 (2260)	<b>0.91</b> (1010)	0.89 (201)
NNP	0.65 (3610)	0.47 (3476)	<b>0.77</b> (330)
XSA	0.68 (1353)	<b>0.71</b> (327)	<b>0.71</b> (142)
VX	0.62 (3624)	0.64 (1451)	<b>0.81</b> (369)
XR	0.41 (826)	<b>0.67</b> (318)	0.49 (52)
NR	0.27 (226)	<b>0.78</b> (73)	0.52 (18)
XPN	0.14 (283)	<b>0.40</b> (83)	0.35 (18)

Table 3: F1 scores (by-tag) in L2 dataset



**Re-evaluation results:** Despite the small size of the training data, the Stanza+L2 model exhibited improvements in the F1 scores of **tokenization (0.93)** and **POS tagging (0.91)** compared to the best models trained exclusively on the L1 dataset (i.e., Stanza, Komoran), which had F1 scores of 0.89 for tokenization and 0.86 for POS tagging. However, when we compared the performance of the three models (i.e., Stanza, Komoran, Stanza+L2) on the L1 dataset, the performance of Stanza+L2 dropped (Token: 0.83; Tag: 0.82). The precise reason for this drop is unclear now; we speculate that it may be an example of "forgetting" (Kirkpatrick et al., 2017) in which neural networks abruptly forget what they have retained when learning a new task. In other words, it may be due to the detailed tagging scheme that our study adopts in comparison to the scheme of the L1 dataset (e.g., parsing causative/passive suffixes). Further research should clarify the interplay between the enhancement of parsing systems and the operation of neural networks in model training.

The by-tag performance of Stanza+L2 (as indicated in the final column of Table 3) shows that the accuracy of 15 out of 29 tags performed better than that for both of the L1 baseline models. The confusion matrix (Figure 2b) further showed that the locus of this improvement was predicate-related tags (VV, VA, VX) and error-related tags (NA, NF, NV). However, for the remaining 17 tags, Komoran still outperformed Stanza+L2. Considering the differences in the pre-training datasets of Stanza and Komoran, the disparity in training data size may have partially accounted for the observed performance discrepancies. Given this context, future research could explore the possibility of expanding Stanza’s L2 training dataset, potentially incorporating a more diverse and comprehensive range of L2-Korean texts to improve its performance in areas in which the Stanza currently trails behind Komoran.

## 6 Conclusion

### 6.1 Summary of findings

In this study, we presented a manually annotated L2-Korean corpus and evaluated the performance of Korean morphological analyzers pre-trained on L1 datasets for tokenization and POS tagging on L2-Korean data. The KLM corpus and related resources are publicly accessible at: <https://github.com/NLPxL2Korean/>

[Korean\\_Learner\\_Morpheme\\_corpus](#).

The results revealed that morphological analyzers exhibited somewhat lower performance on L2-Korean data in comparison to their performance on L1 datasets. A detailed analysis of POS tags showed that several essential morphological tags, including predicate- and suffix-related tags, displayed relatively low accuracy. However, the study demonstrated that substantial improvements in morpheme tokenization and POS tagging performance for L2-Korean data could be attained by incorporating L2 data into the training sets, even with the relatively small dataset. Although no study has specifically focused on L2-Korean data so far, these findings align with previous studies on L2-English UD treebanks (e.g., Berzak et al., 2016; Kyle et al., 2022).

### 6.2 Future directions

To enhance computational resources for lesser-studied languages and improve their performance, carefully designed and validated data-processing pipelines hold great promise. This can be pursued through three primary directions. First, it is essential to expand the size of L2 corpora by (1) refining gold-standard annotation and tagging schemes, and (2) including informative metadata, such as learner proficiency. Second, incorporating syntactic treebanks into the KLM corpus or other available L2-Korean corpora could be considered, as previous research on L2 English has demonstrated promising outcomes. Third, both language-specific properties and learner language characteristics should be taken into account during the resource development process to ensure the interpretability of model results.

### Limitations

Although our study offers empirical reports on the currently available Korean morphological parsers for processing L2-Korean texts, there are remaining areas which await further research. First, the KLM corpus that we proposed in this study consists of a relatively small dataset for training deep-learning models, so increasing the size of the dataset for training may be necessary to fully ensure model performance and generalize the result. Second, the proficiency levels in the original corpus seem unreliable because there was no separate test for proficiency measurement; instead, the developers used class levels as a proxy for learner proficiency.

This invites the need for re-evaluating individual learners' proficiency in Korean, ideally via holistic evaluation of learner essays by human raters. Finally, this work may need larger computing resources when applying cutting-edge deep-learning algorithms, especially with a larger training dataset.

## Ethics Statement

We believe that future research should continue to consider linguistic diversity and give importance to the inclusion of underrepresented languages to research, while promoting equitable research practices in the field. Our findings thus have the potential to contribute to developing more effective and inclusive language-learning resources and tools for language learners. Specifically, connecting the currently available (and L1-based) morphological analyzers to language-specific properties and learner-language characteristics existing in L2 data, including the improvement of their performance, can enhance AI literacy, computer-assisted language learning, and educational materials to meet the unique and individualized needs of language learners with diverse backgrounds.

## Acknowledgments

The authors gratefully acknowledge Youkyung Sung for her contributions to manual annotations and discussions for the enhancement of tokenizing and POS tagging guidelines on the L2 Korean dataset.

## 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.
- Yevgeni Berzak, Jessica Kenney, Carolyn Spadine, Jing Xian Wang, Lucia Lam, Keiko Sophie Mori, Sebastian Garza, and Boris Katz. 2016. [Universal dependencies for learner english.](#) *arXiv preprint arXiv:1605.04278*.
- Yves Bestgen and Sylviane Granger. 2014. [Quantifying the development of phraseological competence in l2 english writing: An automated approach.](#) *Journal of Second Language Writing*, 26:28–41.
- Douglas Biber, Bethany Gray, and Kornwipa Poonpon. 2011. [Should we use characteristics of conversation to measure grammatical complexity in l2 writing development?](#) *Tesol Quarterly*, 45(1):5–35.
- Yige Chen, Eunkyul Leah Jo, Yundong Yao, Kyung-Tae Lim, Miikka Silfverberg, Francis M Tyers, and Jungyeul Park. 2022. [Yet another format of universal dependencies for korean.](#) *arXiv preprint arXiv:2209.09742*.
- S. Cho and Y. Park. 2018. [Characteristics of korean language writing by students at the university of sheffield \(korean:sheffield tayhakkyo hankwuke haksupcauy cakmwun thukseng pwunsek\).](#) *Cakmwun-yenkwu [Korean writing association]*, 38:149–172.
- Jinho D Choi and Martha Palmer. 2011. [Getting the most out of transition-based dependency parsing.](#) In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 687–692.
- Jayeol Chun, Na-Rae Han, Jena D Hwang, and Jinho D Choi. 2018. [Building universal dependency treebanks in korean.](#) In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- Nick C Ellis and Fernando Ferreira-Junior. 2009. [Construction learning as a function of frequency, frequency distribution, and function.](#) *The Modern language journal*, 93(3):370–385.
- Dana Gablasova, Vaclav Brezina, and Tony McEnery. 2017. [Collocations in corpus-based language learning research: Identifying, comparing, and interpreting the evidence.](#) *Language learning*, 67(S1):155–179.
- Janggyoon Jeong, Lee Min-Young, Kwon Minji, and Jung Woo-Sung. 2018. [Classification of writing style by using a morpheme network analysis.](#)
- Boo Kyung Jung. 2022. [The nature of l2 input: Analysis of textbooks for learners of korean as a second language.](#) *Korean Linguistics*, 18(2):182–208.
- JY. Kim, YH. Park, MJ. Kim, HN. Kim, SK. Choi, JH. Suh, and YJ Kwak. 2016. [A study of developing usage searcher of grammar pattern in the korean learner's writing corpus \(korean: hankwuke haksupcauy cakmwun malmwungchilul hwalyonghan mwunhyeng yonglyey kemsaykki kaypal yenkwu\).](#) *Teaching Korean as a Foreign Language*, 44:131–156.
- James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. [Overcoming catastrophic forgetting in neural networks.](#) *Proceedings of the national academy of sciences*, 114(13):3521–3526.
- Kristopher Kyle. 2021. [Natural language processing for learner corpus research.](#) *International Journal of Learner Corpus Research*, 7(1):1–16.
- Kristopher Kyle and Scott Crossley. 2017. [Assessing syntactic sophistication in l2 writing: A usage-based approach.](#) *Language Testing*, 34(4):513–535.



- Kristopher Kyle, Masaki Eguchi, Aaron Miller, and Theodore Sither. 2022. [A dependency treebank of spoken second language english](#). In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 39–45.
- Sun-Hee Lee, M Dickenson, and Ross Israel. 2016. Challenges of learner corpus annotation: Focusing on korean learner language analysis (kolla) system. *Language facts and perspectives*, 38:221–251.
- KyungTae Lim, Jayoung Song, and Jungyeul Park. 2022. [Neural automated writing evaluation for korean l2 writing](#). *Natural Language Engineering*, page 1–23.
- Xiaofei Lu. 2010. [Automatic analysis of syntactic complexity in second language writing](#). *International journal of corpus linguistics*, 15(4):474–496.
- Ryan McDonald, Joakim Nivre, Yvonne Quirnbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, et al. 2013. [Universal dependency annotation for multilingual parsing](#). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97.
- Detmar Meurers and Markus Dickinson. 2017. [Evidence and interpretation in language learning research: Opportunities for collaboration with computational linguistics](#). *Language Learning*, 67(S1):66–95.
- YJ. Nam and UP. Hong. 2014. [Towards a corpus-based approach to korean as a second language \(korean: L2loseuy hankwuke cayenpalhwa khophesuuy kwuchwukkwa hwalyong\)](#). *The Journal of the Humanities for Unification*, 57:193–220.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. [Universal dependencies v2: An evergrowing multilingual treebank collection](#). *arXiv preprint arXiv:2004.10643*.
- Eunjeong L Park and Sungzoon Cho. 2014. [Konlpy: Korean natural language processing in python](#). In *Annual Conference on Human and Language Technology*, pages 133–136. Human and Language Technology.
- Jungyeul Park, Daisuke Kawahara, Sadao Kurohashi, and Key-Sun Choi. 2013. [Towards fully lexicalized dependency parsing for Korean](#). In *Proceedings of the 13th International Conference on Parsing Technologies (IWPT 2013)*, pages 120–126, Nara, Japan. Association for Computational Linguistics.
- Jungyeul Park and Jung Hee Lee. 2016. [A korean learner corpus and its features](#). *En-e-hak [Linguistics]*, (75):69–85.
- Jungyeul Park and Francis Tyers. 2019. [A new annotation scheme for the sejong part-of-speech tagged corpus](#). In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 195–202.
- Karthik Raman, Iftekhar Naim, Jiecao Chen, Kazuma Hashimoto, Kiran Yalasangi, and Krishna Srinivasan. 2022. [Transforming sequence tagging into a seq2seq task](#). *arXiv preprint arXiv:2203.08378*.
- Gyu-Ho Shin and Boo Kyung Jung. 2021. [Automatic analysis of passive constructions in korean: Written production by mandarin-speaking learners of korean](#). *International Journal of Learner Corpus Research*, 7(1):53–82.
- Gyu-Ho Shin and Boo Kyung Jung. 2022. [Input–output relation in second language acquisition: Textbook and learner writing for adult english-speaking beginners of korean](#). *Australian Review of Applied Linguistics*, 45(3):347–370.
- Ho-Min Sohn. 1999. *The Korean language*. New York, NY: Cambridge University Cambridge University Press.

## A Sejong Tag Set

The table provides a Sejong Tag set. The description was sourced from Jeong et al., 2018.

Tag	Description
NNG	Noun, common (보통 명사)
NNP	Proper Noun (고유 명사)
NNB	Noun, common bound (의존 명사)
NR	Numeral (수사)
NP	Pronoun (대명사)
VV	Verb, main (동사)
VA	Adjective (형용사)
VX	Verb, auxiliary (보조 동사)
VCP	Copular, positive (긍정 지정사)
VCN	Copular, negative (부정 지정사)
MM	Determiner (관형사)
MAG	Common adverb (일반 부사)
MAJ	Conjunctive adverb (접속 부사)
IC	Exclamation (감탄사)
JKS	Postposition, nominative (주격 조사)
JKC	Postposition, complement (보격 조사)
JKG	Postposition, pronominal (관형격 조사)
JKO	Postposition, objectival (목적격 조사)
JKB	Postposition, adverbial (부사격 조사)
JKV	Postposition, vocative (호격 조사)
JKQ	Postposition, quotative (인용격 조사)
JC	Postposition, conjunctive (접속 조사)
JX	Postposition, auxiliary (보조사)
EP	Ending, prefinal (선어말 어미)
EF	Ending, closing (종결 어미)
EC	Ending, connecting (연결 어미)
ETN	Ending, nounal (명사형 전성 어미)
ETM	Ending, determinitive (관형형 전성 어미)
XPN	Prefix, nounal (체언 접두사)
XSN	Suffix, verbal (명사 파생 접미사)
XSV	Suffix, verb derivative (동사파생 -)
XSA	Suffix, adjective derivative (형용사 파생 -)
XR	Root (어근)
NF	Undecided (consider a noun) (명사 추정)
NV	Undecided (consider a verb) (용언 추정)
NA	Undecided (분석 불능)
SF	Period, Question, Exclamation (마침표 등)
SE	Ellipsis (줄임표)
SS	Quotation, Bracket, Dash (따옴표 등)
SP	Comma, Colon, Slash (쉼표, 콜론, 빗금)
SO	Hyphen, Swung Dash (붙임표, 물결표)
SW	Symbol (기타기호)
SH	Chinese characters (한자)
SL	Foreign characters (외국어)
SN	Number (숫자)

## B F1 scores (by-tag) in L1 dataset

The table provides the by-tag accuracies from a L1 reference corpus (UD Korean GSD).

Analyzer	Stanza (count)	Komorán (count)
JKO	0.96 <sup>(653)</sup>	0.93 <sup>(246)</sup>
MAJ	0.77 <sup>(44)</sup>	0.68 <sup>(36)</sup>
JKS	0.94 <sup>(564)</sup>	0.95 <sup>(242)</sup>
JKG	0.93 <sup>(323)</sup>	0.94 <sup>(121)</sup>
EF	0.96 <sup>(758)</sup>	0.99 <sup>(328)</sup>
VCN	1.00 <sup>(10)</sup>	1.00 <sup>(3)</sup>
JKB	0.93 <sup>(1005)</sup>	0.91 <sup>(372)</sup>
EC	0.95 <sup>(1590)</sup>	0.94 <sup>(721)</sup>
MAG	0.90 <sup>(622)</sup>	0.95 <sup>(248)</sup>
ETM	0.97 <sup>(967)</sup>	0.92 <sup>(394)</sup>
JX	0.92 <sup>(871)</sup>	0.93 <sup>(382)</sup>
EP	0.94 <sup>(573)</sup>	0.95 <sup>(220)</sup>
NNB	0.91 <sup>(715)</sup>	0.82 <sup>(223)</sup>
XSN	0.88 <sup>(314)</sup>	0.89 <sup>(131)</sup>
ETN	0.82 <sup>(108)</sup>	0.86 <sup>(38)</sup>
NNG	0.91 <sup>(6136)</sup>	0.80 <sup>(1684)</sup>
VCP	0.86 <sup>(334)</sup>	0.90 <sup>(113)</sup>
VV	0.93 <sup>(1478)</sup>	0.88 <sup>(615)</sup>
MM	0.92 <sup>(189)</sup>	0.89 <sup>(78)</sup>
JC	0.85 <sup>(161)</sup>	0.81 <sup>(61)</sup>
XSV	0.93 <sup>(689)</sup>	0.90 <sup>(259)</sup>
VA	0.93 <sup>(458)</sup>	0.96 <sup>(228)</sup>
NP	0.88 <sup>(138)</sup>	0.87 <sup>(71)</sup>
NNP	0.75 <sup>(855)</sup>	0.37 <sup>(793)</sup>
XSA	0.87 <sup>(225)</sup>	0.88 <sup>(90)</sup>
VX	0.91 <sup>(390)</sup>	0.76 <sup>(168)</sup>
XR	0.83 <sup>(206)</sup>	0.94 <sup>(87)</sup>
NR	0.74 <sup>(107)</sup>	0.81 <sup>(28)</sup>
XPN	0.42 <sup>(66)</sup>	0.76 <sup>(14)</sup>

# ChatBack: Investigating Strategies of Providing Synchronous Grammatical Error Feedback in a GUI-based Language Learning Social Chatbot

Kai-Hui Liang<sup>1</sup>, Sam Davidson<sup>2</sup>, Xun Yuan<sup>1</sup>, Shehan Panditharatne<sup>1</sup>, Chun-Yen Chen<sup>3</sup>, Ryan Shea<sup>1</sup>, Derek Pham<sup>1</sup>, Yinghua Tan<sup>3</sup>, Erik Voss<sup>1</sup>, Luke Fryer<sup>4</sup>, Zhou Yu<sup>1,3</sup>

<sup>1</sup>Columbia University, <sup>2</sup>University of California, Davis, <sup>3</sup>Articulate.AI,

<sup>4</sup>The University of Hong Kong,

{kaihui.liang, xy2569, zy2461}@columbia.edu, ssdavidson@ucdavis.edu

## Abstract

The increasing use of AI chatbots as conversational partners for second-language learners highlights the importance of providing effective feedback. To ensure a successful learning experience, it is essential for researchers and practitioners to understand the optimal timing, methods of delivery, and types of feedback that are most beneficial to learners. Synchronous grammar corrective feedback (CF) has been shown to be more effective than asynchronous methods in online writing tasks. Additionally, self-correction by language learners has proven more beneficial than teacher-provided correction, particularly for spoken language skills and non-novice learners. However, existing language-learning AI chatbots often lack synchronous CF and self-correction capabilities. To address this, we propose a synchronous conversational corrective feedback (CCF) method, which allows self-correction and provides metalinguistic explanations (ME). Our experiments examine the effects of different feedback presentation methods and self-correction on users' learning experiences and intention to use the system. Our study suggests that in chatbot-driven language-learning tools, corrective feedback is more effectively delivered through means other than the social chatbot, such as a GUI interface. Furthermore, we found that guided self-correction offers a superior learning experience compared to providing explicit corrections, particularly for learners with high learning motivation or lower linguistic ability.

## 1 Introduction

The growing prevalence of AI chatbots as conversational partners for second-language learners emphasizes the vital role of delivering effective feedback to enhance the overall learning experience. As researchers and practitioners work to optimize computer-based conversational language learning, it is essential to determine the optimal timing, methods of delivery, and feedback types that contribute

to the most successful outcomes. Prior research has shown that synchronous corrective feedback (CF) for grammatical errors is more effective than asynchronous methods in online writing tasks (Shintani and Aubrey, 2016). However, the best form of synchronous CF in AI chatbot systems has yet to be determined. Furthermore, self-correction by language learners has proven to be more beneficial than teacher-provided correction (Brown, 2009), especially for spoken language skills and for learners with more than limited L2 proficiency. Despite this evidence, numerous current language-learning AI chatbots lack diverse synchronous CF and self-correction features. And while past research has shown that learners' proficiency levels significantly influence their preferences (Orts and Salazar, 2016; Yang, 2016; Wiboolyasarin et al., 2022), the optimization of feedback strategies to adapt to users with varying proficiencies and motivations in language-learning chatbots remains unexplored. To address this limitation, we propose a AI chatbot for language learning with synchronous conversational corrective feedback (CCF), and investigate the effect of the feedback form and self-correction with metalinguistic explanations (ME). Specifically, we explore the following two research questions:

**RQ1:** How do the forms of CF delivery, specifically, feedback from the conversational partner (i.e., the chatbot) and a separate role (i.e., a GUI), impact the learning experience, including conversational enjoyment, negative emotions, self-efficacy, perceived usefulness, and intention to use the system? We hypothesize that: **H1:** Learners prefer receiving feedback from a separate role rather than from the conversation partner.

**RQ2:** How does the process of self-correction (compared to explicit feedback without self-correction) impact the learning experiences, including conversational enjoyment, negative emotions, self-efficacy, perceived usefulness, and intention to

use the system? Specifically, what are the effects on people with different linguistic ability and learning purposes? We hypothesize that: **H2.1:** Learners with lower linguistic ability prefer receiving guided self-correction compared to those with higher proficiency. And **H2.2:** Learners with serious learning purposes prefer receiving guided self-correction relative to those who report other learning motivation.

## 2 Related Work

### 2.1 Chatbots as Conversational Partners for L2 Learners

A major challenge for second language instructors and students is finding adequate opportunities for students to practice conversational skills. A possible solution is the use of AI-driven chatbots to fill this gap. For example, [Fryer and Carpenter \(2006\)](#) discuss how chatbots can be used to increase opportunities for students to practice their second language. [Fryer and Carpenter \(2006\)](#) also point out that students who are reticent to speak with human interlocutors are often able to talk more freely with a computer. Similarly, [Huang et al. \(2022\)](#) states that chatbots “encourage students’ social presence by affective, open, and coherent communication.” This interaction is driven by recent advances in generative AI and chatbot design that have improved the dialogue flow of chatbots as well as their adaptability to individual user attributes ([Li et al., 2022](#)). In the present work we combine scripted dialogue with generative AI to create a chatbot which is able to effectively interact with users.

### 2.2 Automatic Corrective Feedback for L2 learners

Providing CF to students is an extremely time-consuming prospect for instructors ([Shintani, 2016](#)), and the automation of feedback can free up instructor time to focus on rhetorical and conversational skills ([Li et al., 2015](#)). Particularly, automated CF (ACF) can provide the type of real-time feedback to students that is impossible for instructors to provide, allowing students to immediately take advantage of the proposed suggestions and gain more confidence in their independent expressive abilities ([Barrot, 2021](#)). [Heift and Hegelheimer \(2017\)](#) further explains that ACF enables “learner self-study and practice of the target language by identifying and explaining error sources” and allows for self-revision.

In the present work, we test two alternate types of CF: explicit and implicit feedback, in the context of an educational chatbot for language learning. Previous work had shown that providing metalinguistic explanations without explicit corrections, which we term guided self-correction, tends to result in better student engagement and immediate gains in target-form usage ([Sauro, 2021](#)) and may improve long-term learning outcomes in writing tasks ([Gao and Ma, 2019](#); [Barrot, 2021](#)). ([Penning de Vries et al., 2020](#)) investigates the use of ACF in a spoken language system, and finds speaking practice with ACF benefits users’ learning goals. However, these feedback methods have not previously been tested in the context of language learning chatbots, a gap that the present paper seeks to address.

An additional key aspect of the present work is our testing alternate strategies for presenting feedback to language learners. Specifically, we test whether students prefer receiving CF directly from the chatbot as part of the conversational flow, or from another source such as the GUI window. While previous work has looked at student reactions to the timing of CF ([Deeva et al., 2021](#)), student control over feedback ([Deeva et al., 2021](#)), and level of explicitness ([Sarré et al., 2021](#); [Sauro, 2021](#)), few studies investigate the effect of method of feedback presentation on engagement and learning experience. As such, this study is the first to investigate the impact of strategies for providing feedback on learning experiences and self-efficacy in the setting of a language learning chatbot.

### 2.3 Grammatical Error Correction & Classification models

Much recent progress has been made in the task of Grammatical Error Correction (GEC). To date, this work has largely focused on student essays ([Ng et al., 2014](#); [Bryant et al., 2019](#)). For example, [Omelianchuk et al. \(2020\)](#)’s GECToR reframes the GEC task as a sequence labeling task rather than a sequence transformation task. Other promising models are proposed by [Stahlberg and Kumar \(2021\)](#) and [Rothe et al. \(2021\)](#), who achieve strong results on the JFLEG ([Napoles et al., 2017](#)) and CoNLL-2014 ([Ng et al., 2014](#)) datasets, respectively. Furthering this work, [Qorib et al. \(2022\)](#) achieves state-of-the-art results on several datasets by combining successful GEC models, such as [Omelianchuk et al. \(2020\)](#) and [Rothe et al. \(2021\)](#)

using a simple logistic regression algorithm. More recently, Fang et al. (2023), Wu et al. (2023), and Coyne and Sakaguchi (2023) have investigated the application of pretrained large language models, such as GPT-3, to GEC benchmark tasks. We emphasize that the above-referenced works primarily target correcting written student essay data. We, on the other hand, seek to apply GEC to the dialogue domain, and thus previously proposed GEC models may not work as effectively as demonstrated in prior art.

The present work also relies on error classification models to ensure that the correct type of feedback is presented to users. ERRANT (Bryant et al., 2017) is a rule-based algorithm to discriminate error categories by their part-of-speech (POS) tags. As an improvement to ERRANT, SERRANT (Choshen et al., 2021) improves the type accuracy by utilizing SErCL (Choshen et al., 2020) rules when ERRANT is not informative. SErCL defines errors by combining the Universal Dependencies (Nivre et al., 2016) tags of the target item before and after correction.

### 3 Study Method

#### 3.1 Recruitment and participants

For this study, we recruited native Mandarin speakers as participants. To find users genuinely interested in conversing with a chatbot and improving their English grammar, we used social media for recruitment, rather than relying on school classes or Amazon Mechanical Turk. Our demographic recruitment criteria included being a native L1 Mandarin speaker aged 18 years or older. We also sought participants having an interest in discussing travel (the topic of the study) in English via text message while receiving grammatical error feedback. Participation in the study was entirely voluntary and unpaid.

175 participants completed the conversation and post-survey, with the following socio-demographic profile. The average age of respondents was 32 years, with the large majority having post-secondary education. Participants have studied English for an average of 15.7 years. Most participants reported self-improvement or having fun as their motivation for engaging with our system. Of those users who participated, 120 users produced one or more targeted errors while using the system. A full breakdown of sociodemographic details can be found in Appendix B.



Figure 1: User study procedure

#### 3.2 Procedure

Figure 1 depicts the user study procedure. Participants were randomly allocated to one of three experimental groups, each implementing a unique grammatical error feedback strategy. The study initiated with a travel-themed conversation with the chatbot. If participants made grammatical errors, as detected by our GEC model, the system offered feedback in accordance with their group’s strategy. To ensure that grammar errors could be identified, users were required to type at least three words per turn and encouraged to use complete sentences. They also needed to complete a minimum of 12 dialogue turns, corresponding to the length of the scripted responses. After the conversation, users completed a post-survey collecting their socio-demographic information, English learning background, motivations, and subjective experiences with the system. To incentivize survey completion, participants who finished the survey received asynchronous grammar feedback, including a conversation summary and grammar error corrections for their responses. Both the system UI and post-survey were in Mandarin.

#### 3.3 Conversation and grammar error feedback

As shown in Figure 2, the conversation alternates between chatting and feedback modes for all experimental groups. It starts with a chatting mode discussing travel with users. Whenever a user makes a grammatical error from the targeted error types (as defined in Section 3.3.1 below), the system first acknowledges their response and then switches to feedback mode. In Group 1, users receive feedback directly from the chatbot (i.e., the interlocutor) via guided self-correction. In Groups 2 and 3, however, users receive feedback via a pop-up window on the system GUI (i.e., separate from the interlocutor) to distinguish it from the conversation. While Group 2 receives guided self-correction, group 3 only receives explicit error correction without an opportunity to self-correct. (See 3.3.2 for more details.) Once the feedback is completed, the system switches back to chatting mode and resumes the ongoing conversation. In case of a non-targeted

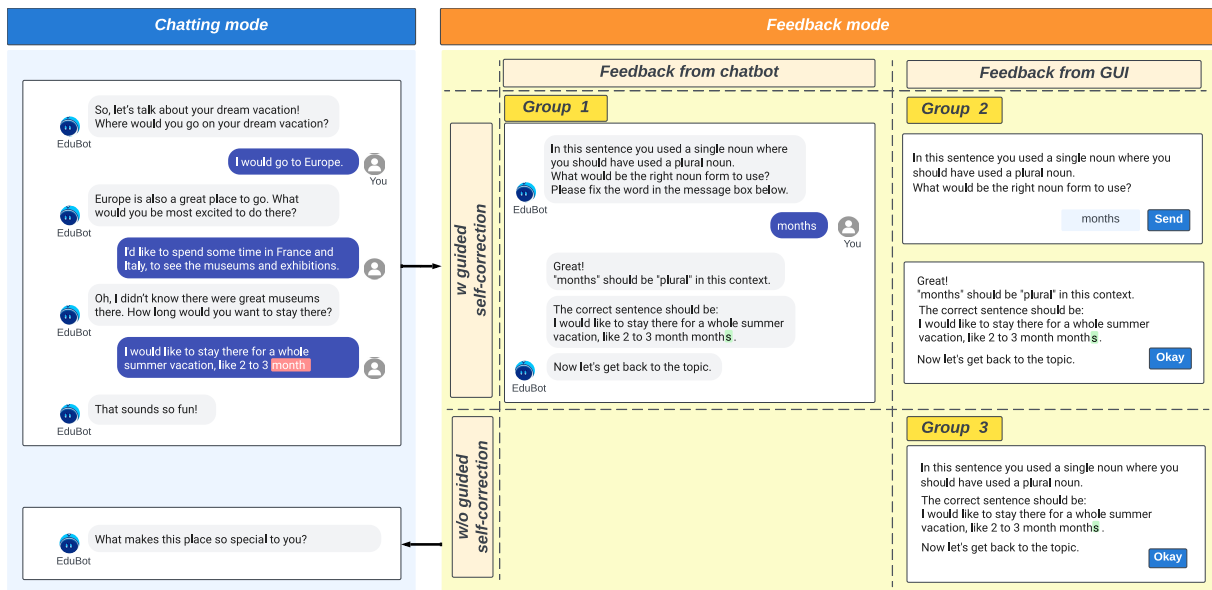


Figure 2: Conversation and feedback flow

error (i.e., an error detected by the GEC model but not explicitly handled by our feedback generator), the system simply highlights the error in the GUI and displays the corrected form at the appropriate location in the user’s previous utterance, without disrupting the chatting mode.

### 3.3.1 Targeted error types

Our current feedback generation method generates feedback for five common types of grammatical errors frequently made by English learners. The error types are defined according to the SERRANT framework (Choshen et al., 2021). The error types we target are as follows:

- **VERB : SVA**: Subject-verb agreement errors.
- **VERB : TENSE**: Incorrect verb tense usage.
- **VERB : FORM**: Verb form errors. For example, using an infinitive verb when a conjugated form is needed.
- **NOUN : NUM**: Noun number errors. For example, a user saying “I like cat” instead of “I like cats”.
- **DET**: Misuse or omission of a determiner, such as “the” or “a”.

We target these errors because they are among the most common errors identified in the ErAConD dataset, indicating a high prevalence of these error types in L2 English learner conversations. We also consulted with professional second language educators who agreed that these error types are among the most frequently seen in their students’ speech.

Finally, to avoid overwhelming students with feedback and disrupting the conversation too frequently, we chose this relatively small set of errors to target for the purposes of this study; we plan to add additional error types in future work.

### 3.3.2 Grammar error feedback strategies

When the user makes a targeted error, we generate CF that includes metalinguistic explanations, hints, and corrected forms. We use the term “metalinguistic” to reference a student’s capacity to “reflect on and manipulate the structural features of language” (Nagy and Anderson, 1995). In the context of the present work, we define “metalinguistic explanation” as feedback which contains explicit information about the student’s language use, such as pointing out that the student used an incorrect verb tense. Depending on the experimental group, the feedback presented to the user can consist of one or more of the following types:

1. Error identification: This specifies the portion of the user’s utterance that contains the error without providing the correct form.
2. Implicit metalinguistic clues: This includes a metalinguistic suggestion about the type of error made, followed by prompts that encourage the user to self-correct, with additional guidance. There are two levels of this type of feedback: Level 1 provides a simple metalinguistic suggestion for the user’s first attempt, while level 2 provides a more detailed metalinguistic explanation for the second attempt.

3. Explicit correction: This provides an explicit statement of the corrected form.

We present these suggestions in different ways depending on the experimental setting. The first type of feedback, which we refer to as *guided self-correction*, begins with feedback types 1 and 2, and progresses to type 3 only if the student is unable to self-correct after two attempts. In this approach, the user is first provided the identified error portion (e.g. “In this sentence you made a mistake on the verb ‘are’.”), along with a metalinguistic suggestion (level 1) and an opportunity to self-correct (e.g. “What verb form should you have used? For example, “sees” and “saw” are different forms of “see.””). If the user is unable to self-correct, they are given a second chance with a more detailed metalinguistic suggestion (level 2) (e.g. “Not quite. Think about subject-verb agreement. How should your verb be changed to agree with the subject “He”?”) If the user is still unable to self-correct after two attempts, we then present the explicit correction containing the corrected form. (e.g. “Good try, but not quite. It’s tricky, I know. The correct verb form here is “is”. Remember to make your verbs agree with their subjects.”) This guided self-correction feedback approach is presented to experimental groups 1 and 2, as shown in Figure 2. The second type of feedback, which we refer to as *explicit feedback*, consists only of providing type 1 and type 3 feedback (see group 3 in Figure 2).

### 3.4 Measurement

#### 3.4.1 Linguistic ability

Linguistic ability includes various aspects. In this study, we focus on learners’ lexical competence in their produced utterances. We measure lexical diversity using the VocD method (McKee et al., 2000)<sup>1</sup> and assess lexical sophistication with the English Vocabulary Profile (EVP), aligning vocabulary usage with CEFR levels. Both metrics are evaluated with the online tool Text Inspector (Bax, 2012), with the medium of text designated as “writing.” While the Text Inspector tool also provides language proficiency levels based on the CEFR framework, we do not rely on this information in our study. The tool’s original design primarily targets writing tasks and may not be as suitable for evaluating language proficiency in textual conversation. For a comprehensive evaluation of the results, please refer to Appendix D.

<sup>1</sup><https://textinspector.com/help/lexical-diversity/>

#### 3.4.2 Post-conversation surveys

Upon the completion of each conversation, we gathered self-reported ratings from users on five distinct constructs related to users’ attitudes toward the system: negative emotion toward the feedback (frustration and annoyance), self-efficacy (confidence in grammar usage and expressive ability), perceived usefulness of the grammatical CF and suggestions, enjoyment using the system, and future intention to use the system. To ensure the reliability and validity of these constructs, we utilized a set of two measurement items, each rated on a 5-point Likert scale, for each construct. These measurement items were adapted from previous research studies (See Table 9) and subsequently modified to better suit the context of language learning chatbots. Figure 5 shows the survey results for each item. Hypotheses related to each construct and detailed descriptions of the constructs are shown in Appendix F.

## 4 System

### 4.1 Overview

Figure 3 presents the system pipeline in chatting mode. At each turn, user input is first processed by the grammar error correction (GEC) module. If any targeted errors are identified, the system switches to feedback mode. The system first highlights the portion of the user’s utterance that contains errors with red backgrounds. Then, the topic chatbot acknowledges the user’s response using its generation model. Subsequently, the conversational feedback generator provides grammatical feedback to the users. The feedback content and form of delivery will vary depending on the group’s feedback strategies. For non-targeted error types, the topic chatbot will continue the conversation while the system will highlight the user’s error and display the corrected form on the GUI at the user’s previous response. If there are no grammar errors in the user’s input, the topic chatbot continues the conversation without highlighting or interruption.

The process in feedback mode, where targeted types are being addressed, proceeds as follows: For the group without guided self-correction (group 3), the system switches back to chatting mode immediately after providing explicit grammatical feedback at the same turn. For groups with guided self-correction (groups 1 and 2), the feedback mode continues to the next turn until the correction process concludes. During feedback mode in subsequent turns, the GEC module checks if users are able

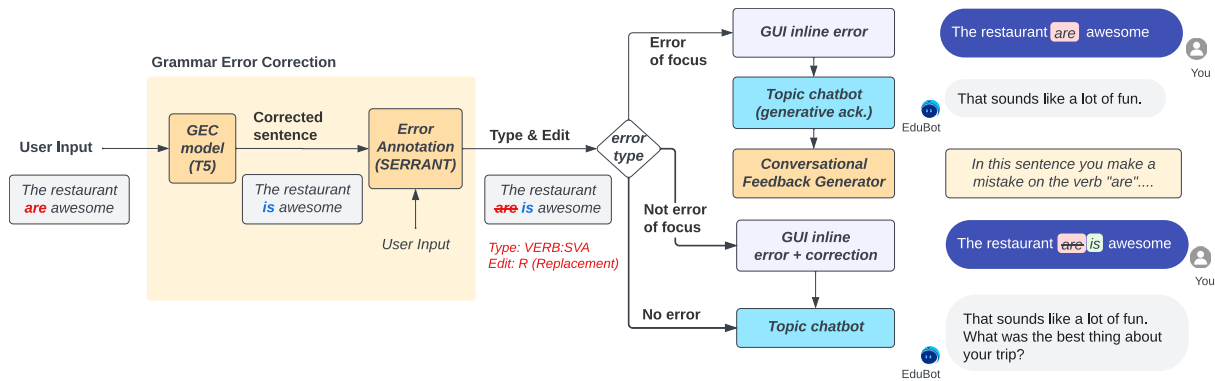


Figure 3: System pipeline in chatting mode: Grammar error correction & response generation flow

to successfully self-correct their errors. If users self-correct successfully, the feedback generator acknowledges the correction and the system returns to chatting mode where the topic chatbot continues the conversation. If they don't, they are given a second chance where the feedback generator provides a more detailed metalinguistic hint. If they fail to self-correct after two attempts, the feedback generator provides explicit feedback the system switches back to chatting mode. Otherwise, the feedback continues.

## 4.2 Topic chatbot

The topic chatbot combines scripted dialogue with a generative model to create a topic-oriented chatbot capable of effectively interacting with users. At every dialogue turn, the chatbot first generates a response and subsequently concatenates it with the scripted responses. Scripted dialogue is employed for experimental control purposes, primarily to pose questions designed to elicit more grammatical errors and to ensure consistency in the topics presented to users across different experimental groups. Conversely, the generative model is used to acknowledge user responses in a more natural manner by dynamically responding to user input.

The script encompasses 12 dialogue turns covering travel preferences, past travel experiences, and dream vacations. We employ Blenderbot3 3B as our generative model, which possesses various conversational skills and long-term memory. To reduce latency, Blenderbot's internet access was disabled during experiments. After completing the scripted portion of the conversation, if users decide to continue the conversation, the chatbot's responses will rely solely on the generative model.

## 4.3 Grammatical Error Feedback

### 4.3.1 Grammar error correction

Table 1: Performance of GEC model. TP, FP and FN denote the average number of true positives, false positives and false negatives among 5 runs of cross-validation, respectively.

Model	TP	FP	FN	Prec	Rec	$F_{0.5}$
GECToR	24.6	14.4	174.0	0.63	0.12	0.34
T5 (Ours)	43.8	34.6	154.8	0.56	0.23	0.43

Figure 3 illustrates the grammar error correction process, which consists of two main steps: grammar error correction and error annotation. First, we use a grammar error correction (GEC) model to generate corrected sentences based on user-input sentences. The GEC model is a T5 (Raffel et al., 2020) model trained for grammar correction<sup>2</sup>. We fine-tuned the model on the ErAConD dataset (Yuan et al., 2022), a GEC conversation dataset between L2 English learners (of at least intermediate proficiency level) and an educational chatbot. We selected level 3 errors (as defined in the ErAConD dataset) as our training data since they are most likely to result in misunderstanding. The resulting fine-tuned model achieves an overall  $F_{0.5}$  of 0.43 evaluated by 5-fold cross-validation, as shown in Table 1. Detailed results by error type are shown in Appendix Table 10. While our reported  $F_{0.5}$  is substantially lower than SOTA GEC models designed for written text, there is no established baseline for dialog GEC. Note that the precision of 0.56 doesn't mean that half of the edits generated are incorrect. In fact, there are many equally valid ways to correct a given grammar error; however, when

<sup>2</sup><https://huggingface.co/deep-learning-analytics/GrammarCorrector>



calculating precision using a test dataset, we can only compare system-generated corrections with the one or two human-annotated gold edits. If the machine-generated correction does not match the gold annotation, it will negatively impact evaluation performance, even if the correction is a completely legitimate alternative. As a result, current evaluations tend to underestimate the performance of GEC models. [Rozovskaya and Roth \(2021\)](#) provides an in-depth study of this issue. While the current model is effective for the present study, we are working to improve the GEC model for future iterations of our system.

After error correction by the GEC model, SERRANT compares the user input sentence with the corrected version to extract edits and classify error types. For most categories, there are three possible operations to specify user input errors: Missing (M), Replacement (R), and Unnecessary (U), indicating whether tokens should be inserted, substituted, or removed, respectively. Subsequently, we filter out trivial grammar error types (e.g., punctuation) and reapply the edits to the original sentences.

### 4.3.2 Grammar error feedback presentation

Grammar errors can be presented in three different forms: 1) GUI inline highlighting on the user’s utterance, 2) conversational feedback presented in the form of a chatbot response from the feedback generation module, and 3) conversational feedback presented in a pop-up window from the feedback generation module.

As discussed in Section 3.3.1, our feedback generation module explicitly targets five error types, while other error types detected by our GEC model are referred to as “non-targeted”. For targeted errors, the error is first presented in the form of GUI inline highlighting on the user’s previous response. Then, after the topic chatbot acknowledges the user’s content, conversational feedback is presented in a form that depends on the experiment group. For group 1, the feedback is presented by the chatbot, while for groups 2 and 3, it is presented in a pop-up window. For non-targeted errors, only GUI inline highlighting is shown without any additional feedback.

To generate conversational feedback, we rely on a number of feedback templates that can be modified based on the specifics of the respective error. For example, if SERRANT tags an error as `R:NOUN:NUM`, indicating a replacement operation (‘R’) resulting from a difference in noun num-

ber between the original input and the correction, we populate a template with noun number information to generate feedback such as “In this sentence, you used a single noun when you should have used a plural noun”, as shown in Figure 2. We use a similar approach to populate feedback templates for error types such as subject-verb agreement, verb tense, verb forms, and determiners.

## 5 Results

### 5.1 Dialog statistics

Table 2 displays the distribution of participants across each experimental group. Among the 175 participants, 154 encountered at least one error, with 120 experiencing at least one targeted error. In this study, our survey analysis focuses on the 120 users who encountered targeted errors, since the primary experimental treatment involved the feedback delivery strategy for these errors.

Table 3 offers statistics for users who had targeted errors in their conversations, with a sample size of 120. On average, users engaged in 15.1 dialog turns (i.e. 15.1 responses from users), each consisting of 10.1 tokens. Each conversation contained 3.4 turns with any error, 1.6 turns with non-targeted errors exclusively, and 1.8 turns with targeted errors. The average number of errors per dialog amounted to 4.3. We also analyzed the most frequently occurred error types among all 175 participants, with the top ten including the five targeted error types as well as preposition, spelling, noun, and verb errors (see Appendix E for comprehensive error type counts).

Regarding learners’ lexical competence, we assessed their lexical diversity, which had a mean (M) value of 84.8 (SD = 27.0) and a median of 80.25. The range of lexical diversity scores ranged from 37.1 to 200 (see Appendix D for more details).

Table 2: Numbers of participants in each group

Group	All	W/ any err.	W/ targeted err.
Group 1	49	43	33
Group 2	66	60	48
Group 3	60	51	39
<b>Total</b>	<b>175</b>	<b>154</b>	<b>120</b>

### 5.2 Survey results

Figure 5 Shows the survey results of all dialogs with targeted errors. We performed two-tailed t-tests between groups (Groups 1 and 2 for RQ1,

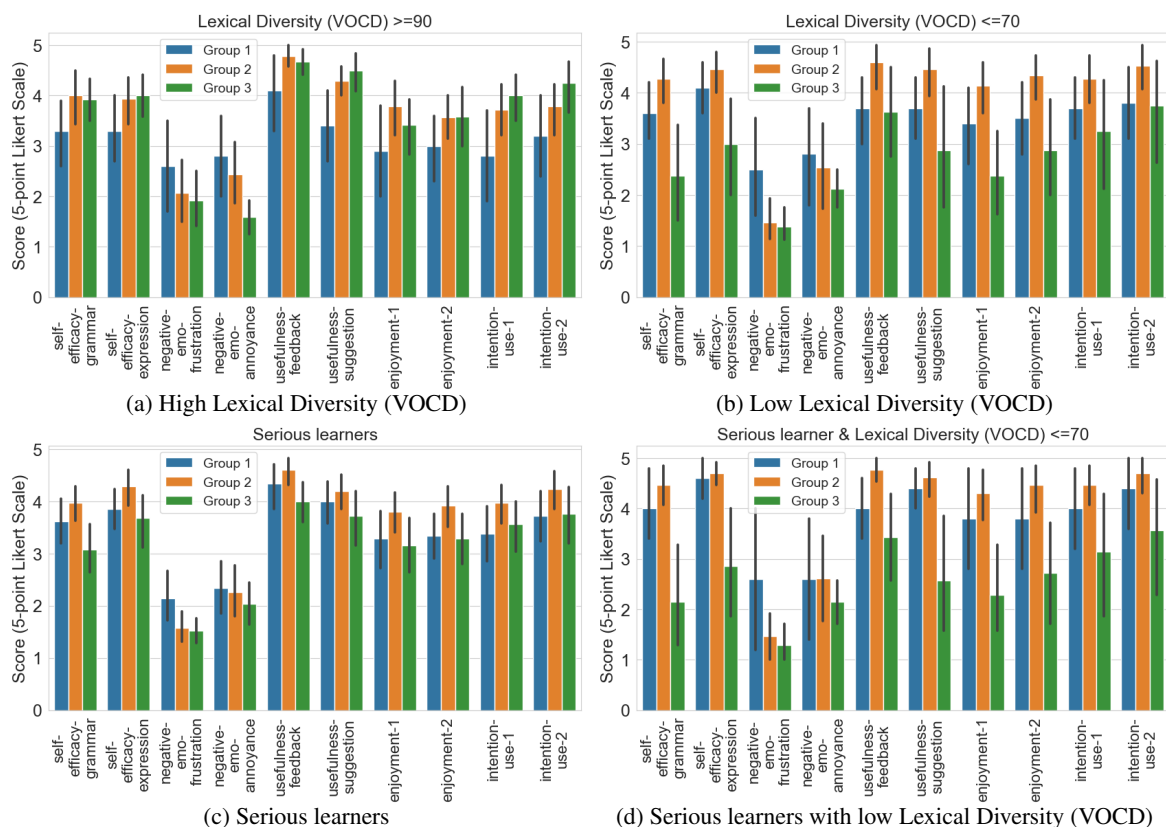


Figure 4: Survey results of learners with with different lexical diversities and motivation.

Table 3: Dialog statistics

Item	M ± SD	Mdn.	Range
# of dialog turns	15.1 ± 5.2	13	13-47
# of tokens per turn	10.1 ± 4.4	9	4-29
# of turns w/ any error	3.4 ± 2.2	3	1-16
# of turns w/ non-targeted errors only	1.6 ± 1.7	1	0-10
# of turns w/ targeted error	1.8 ± 1.0	1	1-6
# of errors per dialog	4.3 ± 3.6	3	1-31

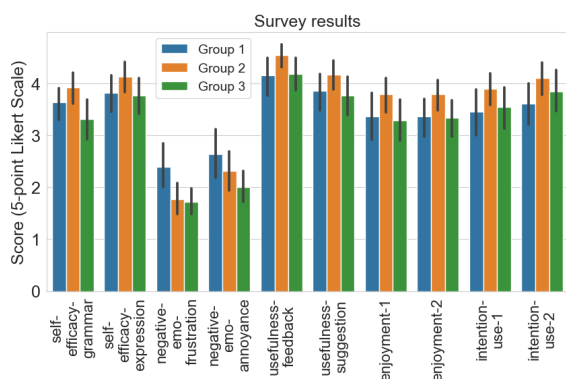


Figure 5: Survey results

and Groups 2 and 3 for RQ2), and use Welch t-

test when the sample sizes are unequal, as recommended by Zimmerman (2004).

### 5.2.1 Effects of the form of feedback delivery

The results presented in Figure 5 demonstrate that users experienced higher frustration levels when interacting with Group 1 than with Group 2 ( $t(58.61) = 2.26, p < .05$ ). Our findings suggest that feedback provided by the dialogue agent leads to greater frustration than feedback delivered from another role, such as the GUI, even when the content and timing of the feedback are identical.

### 5.2.2 Effects of guided self-direction

Figure 5 shows that users gained more self-efficacy in their grammar skills when interacting with Group 2 compared to Group 3 ( $t(77.88) = 2.51, p < .05$ ). These results suggest that guided self-correction may be beneficial for enhancing users' confidence in their English grammar skills during conversations.

**Effects of user's linguistic ability** To examine the influence of guided self-correction on users with varying linguistic abilities, we analyzed survey data from participants with higher and lower lexical diversities (VocD  $\geq 90$  and VocD  $\leq 70$ ,

respectively). The threshold values were determined based on the median VocD score (80) with a range of plus or minus 10. Our results indicate that users with higher lexical diversity found guided self-correction (Group 2) more annoying compared to the absence of guided self-correction (Group 3). This could be because users with higher lexical competence might have already understood the corresponding metalinguistic rules, making guided self-correction redundant and less efficient than explicit feedback.

**Effects on users' motivation** To investigate the effects on users with varying motivations, particularly their level of commitment to improving their English conversation skills, we excluded approximately one-third of users who reported using the system out of curiosity or for fun and defined the remaining users as "serious learners". Our findings (Figure 4c) reveal that serious learners not only experienced significantly higher levels of confidence in their grammar skills with guided self-correction ( $t(46.57) = 2.96, p < .01$ ), but also perceived the feedback to be more useful compared to the absence of guided self-correction ( $t(40.54) = 2.47, p < .01$ ). Moreover, we conducted a further analysis on serious learners with low lexical diversity ( $\text{VOC} \leq 70$ ) (Figure 4d) and found that when receiving guided self-correction, they reported higher enjoyment in conversation ( $t(9.14) = 3.46, p < .01$  for enjoyment-1 and  $t(8.28) = 2.84, p < .05$  for enjoyment-2), increased self-efficacy in both grammar skills ( $t(8.21) = 4.20, p < .01$ ) and expressing ideas ( $t(6.61) = 3.01, p < .05$ ), and perceived the grammatical corrective feedback ( $t(6.78) = 2.70, p < .05$ ) and suggestions ( $t(6.94) = 3.03, p < .05$ ) to be more useful compared to the absence of guided self-correction.

## 6 Conclusion

Results from this preliminary study provide evidence that learners may prefer getting corrective feedback from a separate role, instead of from the conversation partner to reduce frustration. In addition, guided self-correction may provide better learning experiences than the absence of self-correction, especially for learners with lower lexical competence or more serious learning motivation. These findings highlight the importance of considering users' individual differences when designing language-learning chatbots, and the need

for personalized feedback mechanisms that cater to individual users' need.

## 7 Limitations

### 7.1 Assessment of learner's linguistic ability and future research

In this study, the assessment of learners' linguistic ability was limited to analyzing the learners' produced utterances in a single short conversation. Also, it was analyzed with the online tool TextInspector, which was primarily designed for evaluating writing tasks rather than textual conversation. While this provides some insight into their language proficiency, a more comprehensive assessment of learners' language proficiency could offer a deeper understanding of how it influences their preference toward different feedback strategies. Future research should consider incorporating additional measures to evaluate learners' language proficiency comprehensively. This could involve utilizing standardized tests for receptive and productive skills and conducting detailed assessments of vocabulary, grammar, and discourse abilities.

### 7.2 Effect of participants' language proficiency

In this study, survey data were collected from participants capable of engaging in a conversation about travel with at least 12 turns from each side. Participants without the ability to meet this requirement were automatically excluded and did not complete the post-survey. Previous research (Van Beuningen et al., 2012) indicates that learners with limited proficiency may prefer explicit corrective feedback, as they may face challenges in independently arriving at correct answers. However, it should be noted that due to the inherent study design, some learners with limited proficiency might not have been included in the sample.

### 7.3 Effect of the GEC model performance

During the experiment, there were no existing GEC (Grammar Error Correction) models specifically designed for conversational grammar errors. As a result, we developed our own GEC model using a small dataset of GEC dialogues. To enhance the performance of the GEC model in future iterations, we are actively working on collecting additional conversational GEC datasets. By incorporating more diverse and extensive data, we aim to improve the accuracy and effectiveness of the GEC

model. The enhanced performance of the GEC model is anticipated to have an impact on the effectiveness of different feedback strategies. A more proficient GEC model could potentially yield better user experiences, resulting in higher intentions to use the system. The availability of improved GEC capabilities will enable more precise and tailored feedback, enhancing the overall effectiveness of the system.

#### 7.4 Effect of different feedback strategies

In this study, all feedback strategies used were interruptive, potentially disrupting the conversation flow. However, learners with higher linguistic ability may prefer fewer interruptions, such as preferring no self-correction than self-correction. Additionally, it is important to acknowledge that individual learners may have different preferences and learning styles. To address this, future systems could consider non-intrusive feedback strategies. For example, grammar errors could be highlighted with a background color, and optional metalinguistic explanations could be provided on-demand. This allows learners to access guidance without forcefully interrupting the conversation, catering to their preferences and maintaining a smoother learning experience.

#### References

- Martínez Agudo and Juan de Dios. 2013. An investigation into how efl learners emotionally respond to teachers' oral corrective feedback. *Colombian Applied Linguistics Journal*, 15(2):265–278.
- Jessie S Barrot. 2021. Using automated written corrective feedback in the writing classrooms: effects on l2 writing accuracy. *Computer Assisted Language Learning*, pages 1–24.
- S Bax. 2012. Text inspector. *Online text analysis tool*.
- Alan V Brown. 2009. Students' and teachers' perceptions of effective foreign language teaching: A comparison of ideals. *The modern language journal*, 93(1):46–60.
- Christopher Bryant, Mariano Felice, Øistein E Andersen, and Ted Briscoe. 2019. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75.
- Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805.
- Leshem Choshen, Dmitry Nikolaev, Yevgeni Berzak, and Omri Abend. 2020. Classifying syntactic errors in learner language. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 97–107.
- Leshem Choshen, Matanel Oren, Dmitry Nikolaev, and Omri Abend. 2021. [SERRANT: a syntactic classifier for english grammatical error types](#). *CoRR*, abs/2104.02310.
- Steven Coyne and Keisuke Sakaguchi. 2023. An analysis of gpt-3's performance in grammatical error correction. *arXiv preprint arXiv:2303.14342*.
- Galina Deeva, Daria Bogdanova, Estefanía Serral, Monique Snoeck, and Jochen De Weerd. 2021. A review of automated feedback systems for learners: Classification framework, challenges and opportunities. *Computers & Education*, 162:104094.
- Tao Fang, Shu Yang, Kaixin Lan, Derek F Wong, Jinpeng Hu, Lidia S Chao, and Yue Zhang. 2023. Is chatgpt a highly fluent grammatical error correction system? a comprehensive evaluation. *arXiv preprint arXiv:2304.01746*.
- Luke Fryer and Rollo Carpenter. 2006. Bots as language learning tools. *Language Learning & Technology*, 10(3):8–14.
- Jianwu Gao and Shuang Ma. 2019. The effect of two forms of computer-automated metalinguistic corrective feedback.
- Trude Heift and Volker Hegelheimer. 2017. Computer-assisted corrective feedback and language learning. *Corrective feedback in second language teaching and learning*, pages 51–65.
- Weijiao Huang, Khe Foon Hew, and Luke K Fryer. 2022. Chatbots for language learning—are they really useful? a systematic review of chatbot-supported language learning. *Journal of Computer Assisted Learning*, 38(1):237–257.
- Jinrong Li, Stephanie Link, and Volker Hegelheimer. 2015. [Rethinking the role of automated writing evaluation \(AWE\) feedback in ESL writing instruction](#). *Journal of Second Language Writing*, 27:1–18.
- Yu Li, Chun-Yen Chen, Dian Yu, Sam Davidson, Ryan Hou, Xun Yuan, Yinghua Tan, Derek Pham, and Zhou Yu. 2022. Using chatbots to teach languages. In *Proceedings of the Ninth ACM Conference on Learning@ Scale*, pages 451–455.
- Mingxu Liu. 2013. English bar as a venue to boost students' speaking self-efficacy at the tertiary level. *English Language Teaching*, 6(12):27–37.

- Gerard McKee, David Malvern, and Brian Richards. 2000. Measuring vocabulary diversity using dedicated software. *Literary and linguistic computing*, 15(3):323–338.
- William E Nagy and Richard C Anderson. 1995. Metalinguistic awareness and literacy acquisition in different languages. technical report no. 618.
- Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. Jfleg: A fluency corpus and benchmark for grammatical error correction. *arXiv preprint arXiv:1702.04066*.
- Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14.
- Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666.
- Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhashnyi. 2020. GECToR—Grammatical Error Correction: Tag, Not Rewrite. *arXiv preprint arXiv:2005.12592*.
- Sara Orts and Patricia Salazar. 2016. Efl students’ preferences towards written corrective feedback: An exploratory study on age and level of proficiency. *The Grove-Working Papers on English Studies*, 23.
- Bart WF Penning de Vries, Catia Cucchiari, Helmer Strik, and Roeland Van Hout. 2020. Spoken grammar practice in call: The effect of corrective feedback and education level in adult l2 learning. *Language Teaching Research*, 24(5):714–735.
- Muhammad Qorib, Seung-Hoon Na, and Hwee Tou Ng. 2022. Frustratingly easy system combination for grammatical error correction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1964–1974.
- 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. *The Journal of Machine Learning Research*, 21(1):5485–5551.
- Sonny Rosenthal and Rabindra A Ratan. 2022. Balancing learning and enjoyment in serious games: Kerbal space program and the communication mediation model. *Computers & Education*, 182:104480.
- Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. A simple recipe for multilingual grammatical error correction. *arXiv preprint arXiv:2106.03830*.
- Alla Rozovskaya and Dan Roth. 2021. How good (really) are grammatical error correction systems? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2686–2698.
- Tracii Ryan and Michael Henderson. 2018. Feeling feedback: students’ emotional responses to educator feedback. *Assessment & Evaluation in Higher Education*, 43(6):880–892.
- Raafat George Saadé, Weiwei Tan, and Fassil Nebebe. 2008. Impact of motivation on intentions in online learning: Canada vs china. *Issues in Informing Science & Information Technology*, 5.
- Cédric Sarré, Muriel Grosbois, and Cédric Brudermann. 2021. Fostering accuracy in l2 writing: Impact of different types of corrective feedback in an experimental blended learning efl course. *Computer Assisted Language Learning*, 34(5-6):707–729.
- Shannon Sauro. 2021. Computer-mediated corrective feedback and the development of l2 grammar. *UMBC Education Department Collection*.
- Natsuko Shintani. 2016. [The effects of computer-mediated synchronous and asynchronous direct corrective feedback on writing: a case study](#). *Computer Assisted Language Learning*, 29(3):517–538.
- Natsuko Shintani and Scott Aubrey. 2016. The effectiveness of synchronous and asynchronous written corrective feedback on grammatical accuracy in a computer-mediated environment. *The Modern Language Journal*, 100(1):296–319.
- Felix Stahlberg and Shankar Kumar. 2021. Synthetic data generation for grammatical error correction with tagged corruption models. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 37–47.
- Ting Sun and Chuang Wang. 2020. College students’ writing self-efficacy and writing self-regulated learning strategies in learning english as a foreign language. *System*, 90:102221.
- Catherine G Van Beuningen, Nivja H De Jong, and Folkert Kuiken. 2012. Evidence on the effectiveness of comprehensive error correction in second language writing. *Language learning*, 62(1):1–41.
- Kanokpan Wiboolyasarin, Ruedee Kamonsawad, Natatwat Jinowat, and Watcharapol Wiboolyasarin. 2022. Efl learners’ preference for corrective feedback strategies in relation to their self-perceived levels of proficiency. *English Language Teaching Educational Journal*, 5(1):32–47.

Haoran Wu, Wenxuan Wang, Yuxuan Wan, Wenxiang Jiao, and Michael Lyu. 2023. Chatgpt or grammarly? evaluating chatgpt on grammatical error correction benchmark. *arXiv preprint arXiv:2303.13648*.

Juan Yang. 2016. Learners' oral corrective feedback preferences in relation to their cultural background, proficiency level and types of error. *System*, 61:75–86.

Xun Yuan, Derek Pham, Sam Davidson, and Zhou Yu. 2022. Eracond: Error annotated conversational dialog dataset for grammatical error correction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 76–84.

Donald W Zimmerman. 2004. A note on preliminary tests of equality of variances. *British Journal of Mathematical and Statistical Psychology*, 57(1):173–181.

## A Supplementary Materials

The detailed experiment results related to this paper are available in the following GitHub repository: [https://github.com/KaihuiLiang/chatback\\_gec\\_feedback](https://github.com/KaihuiLiang/chatback_gec_feedback)

In the following sections, we have selected the most critical aspects of these results for a concise understanding.

## B Sociodemographics of participants

Table 4: Sociodemographics

	All users (N=175)			Users with targeted errors (N=120)		
Sociodemographics	n (%) or M ± SD	Mdn.	Range	n (%) or M ± SD	Mdn.	Range
<b>Age (years)</b>	32.0 ± 13.7	26	18-70	32.1 ± 13.2	26.5	18-70
<b>Gender</b>						
Women	99 (56.6%)			73 (60.8%)		
Men	66 (37.7%)			38 (31.7%)		
Prefer not to say	10 (5.7%)			9 (7.5%)		
<b>Education</b>						
Graduate	90 (51.4%)			61 (50.8%)		
Undegraduate	73 (41.7%)			52 (43.3%)		
High school	9 (5.1%)			5 (4.2%)		
others	3 (1.7%)			2 (1.7%)		
<b>Motivation</b>						
Self improvement	69 (39.4%)			50 (41.7%)		
For fun	62 (35.4%)			39 (32.5%)		
Pass tests	15 (8.6%)			12 (10.0%)		
others	12 (6.9%)			10 (8.3%)		
Talk to friends/families	5 (2.9%)			3 (2.5%)		
Travel	5 (2.9%)			3 (2.5%)		
Learn cultures	4 (2.3%)			2 (1.7%)		
Job opportunities	3 (1.7%)			1 (0.8%)		
<b>Learning duration</b>	15.7 ± 9.9	14	0-55	16 ± 9.4	15	0-50

## C Dialog statistics and grammar error counts

Table 5: Dialog statistics

Dialog stats.	All users (N=175)			Users w/ targeted err. (N=120)		
Item	M ± SD	Mdn.	Range	M ± SD	Mdn.	Range
# of dialog turns	14.8 ± 4.6	13	13-47	15.1 ± 5.2	13	13-47
# of tokens per turn	9.8 ± 4.4	9	3-31	10.1 ± 4.4	9	4-29
# of turns w/ any error	2.7 ± 2.3	1	0-6	3.4 ± 2.2	3	1-16
# of turns w/ non-targeted errors only	1.5 ± 1.7	1	0-10	1.6 ± 1.7	1	0-10
# of turns w/ targeted error	1.2 ± 1.2	2	0-16	1.8 ± 1.0	1	1-6
# of errors per dialog	3.4 ± 3.5	3	0-31	4.3 ± 3.6	3	1-31

## D Participants' lexical competence and language proficiency levels

Table 6: Users' lexical competence. All scores are measured by TextInspector based on users' responses.

	All users (N=175)			Users with targeted errors (N=120)		
Lexical competence	M ± SD	Mdn.	Range	M ± SD	Mdn.	Range
<b>Lexical Diversity</b>						
VocD	81.8 ± 27.8	78.5	0-200	84.8 ± 27.0	80.25	37.1 - 200
MTLD	76.8 ± 27.5	73.6	0-176.4	78.8 ± 25.9	74.7	30.1-176.4
<b>Lexical Sophistication: English Vocabulary Profile (EVP)</b>						
C2 type %	0.3 ± 0.6	0	0-2	0.3 ± 0.6	0	0-2
C1 type %	0.5 ± 0.7	0	0-4	0.5 ± 0.7	0	0-3
B2 type %	2.1 ± 1.9	1.7	0-8	2.0 ± 1.8	1.7	0-6
B1 type %	7.2 ± 3.2	6.8	0-16	7.1 ± 3.3	6.7	0-16
A2 type %	15.4 ± 4.5	15	5-30	15.8 ± 4.6	15.5	7-30
A1 type %	63.3 ± 6.6	63.4	46-80	63.0 ± 6.5	63.2	47-80
C2 token %	0.2 ± 0.4	0	0-2	0.2 ± 0.4	0	0-2
C1 token %	0.3 ± 0.5	0	0-3	0.4 ± 0.5	0	0-2
B2 token %	1.5 ± 1.3	1.2	0-5	1.4 ± 1.2	1.3	0-5
B1 token %	5.2 ± 2.4	5	0-11	5.2 ± 2.5	5	0-11
A2 token %	11.8 ± 3.4	11.4	5-23	12.0 ± 3.4	11.5	6-23
A1 token %	71.9 ± 5.4	72	53-85	71.9 ± 5.2	71.9	53-85

Table 7: Users' language proficiency levels. All scores are measured by TextInspector based on users' responses. The overall CEFR represents the holistic score derived from all available metrics. The "VocD - CEFR level" indicates the CEFR level determined by the VocD score, while the "MTLD - CEFR level" represents the CEFR level determined by the MTLD score.

	Overall CEFR level		VocD - CEFR level		MTLD - CEFR level	
Level	All users (N=175)	Users with targeted err. (N=120)	All users (N=175)	Users with targeted err. (N=120)	All users (N=175)	Users with targeted err. (N=120)
<b>C2</b>	1 (0.6%)	0	0	0	0	0
<b>C1+</b>	0	0	0	0	7 (4.0%)	6 (5.0%)
<b>C1</b>	4 (2.3%)	4 (3.3%)	0	0	5 (2.9%)	4 (3.3%)
<b>B2+</b>	26 (14.9%)	19 (15.8%)	18 (10.3%)	16 (13.3%)	9 (5.1%)	6 (5.0%)
<b>B2</b>	47 (26.9%)	31 (25.8%)	23 (13.1%)	16 (13.3%)	12 (6.9%)	10 (8.3%)
<b>B1+</b>	56 (32.0%)	38 (31.7%)	23 (13.1%)	17 (14.2%)	17 (9.7%)	13 (10.8%)
<b>B1</b>	32 (18.3%)	24 (20.0%)	12 (6.9%)	7 (5.8%)	12 (6.9%)	7 (5.8%)
<b>A2+</b>	7 (4.0%)	3 (2.5%)	0	0	0	0
<b>A2</b>	2 (1.1%)	1 (0.8%)	29 (16.6%)	18 (15.0%)	0	0
<b>A1</b>	0	0	0	0	38 (21.7%)	25 (20.8%)
<b>N/A</b>	0	0	70 (40.0%)	46 (38.3%)	75 (42.9%)	49 (40.8%)



## E Grammar error type counts

Table 8: Grammar error type counts in utterances of all participants. Targeted errors are highlighted with a yellow background. "op." denotes operations: R for Replacement, M for Missing, U for Unnecessary. The error types are defined according to the SERRANT framework (Choshen et al., 2021).

Error type (w/ op.)	Count	%	Error type (w/o op.)	Count	%
R:NOUN:NUM	70	11.7	PREP	71	11.9
R:SPELL	62	10.4	NOUN:NUM	70	11.7
R:VERB:FORM	47	7.9	DET	62	10.4
R:VERB:SVA	38	6.4	SPELL	62	10.4
M:DET	38	6.4	VERB:FORM	60	10
R:PREP:WC	34	5.7	VERB:SVA	38	6.4
M:PREP	20	3.3	NOUN	34	5.7
R:OTHER	20	3.3	VERB:TENSE	22	3.7
R:VERB:TENSE	17	2.8	VERB	21	3.5
R:NOUN:WC	16	2.7	OTHER	20	3.3
U:DET	15	2.5	OTHER:MW	14	2.3
U:PREP	15	2.5	PRON	11	1.8
R:OTHER:MW	14	2.3	AUX:MW	9	1.5
U:NOUN	14	2.3	VERB:MW	8	1.3
M:VERB:FORM	12	2.0	NOUN->VERB	7	1.2
R:DET:WC	9	1.5	VERB:INFL	5	0.8
R:AUX:MW	9	1.5	NOUN:INFL	5	0.8
R:VERB:WC	9	1.5	NOUN->PRON	4	0.7
R:VERB:MW	8	1.3	ADV	4	0.7
R:NOUN->VERB	7	1.2	ADJ	4	0.7

## F Survey constructs

Table 9 shows all survey questions and references.

**Negative emotions** For negative emotions towards feedback, we measured users' negative emotions, specifically their levels of frustration and annoyance when receiving immediate corrections during the conversation. Our hypotheses were that users would experience fewer negative emotions in two scenarios: 1) when receiving corrections from the GUI, which is a separate role from the chatbot; and 2) when not required to correct themselves.

**Self-efficacy** Regarding self-efficacy, we measured the level of self-efficacy that users gained after the conversation, specifically their confidence in their grammar skills and their ability to express ideas in English conversations. Our hypotheses were that users would experience a greater increase in self-efficacy when: 1) corrections were given through the GUI, which would provide a less frustrating experience; and 2) they were given the opportunity for guided self-correction, allowing them to actively participate in the learning process and gain a better understanding of their mistakes.

**Usefulness** For usefulness, we measured the level of perceived usefulness of the grammatical CF by users. Our hypothesis was that guided self-correction would be perceived as more useful than without.

**Enjoyment** Regarding enjoyment, we measured the level of enjoyment that users experienced while conversing with the chatbot. Our hypothesis was that receiving grammatical correction feedback from the GUI would be more enjoyable than from the chatbot, as the interruptive feedback would be given from a separate role rather than the conversation partner. Additionally, we hypothesized that higher proficiency

learners would find having a conversation without guided self-correction more enjoyable, as they would require less self-correction and experience fewer interruptions.

**Intention to use** Lastly, we asked users if they intended to use the system again, using one item that was reverse-coded for a sanity check. Our hypothesis was that users would have a higher intention to use the system if they experienced less negative emotion, gained more self-efficacy, perceived the system as more useful, and enjoyed the conversation more.

Table 9: Survey questions

Construct	Item abbr.	Question	Reference
Self-efficacy	self-efficacy-grammar	I think my grammar skills in English conversations improved after using the system	(Sun and Wang, 2020)
	self-efficacy-expression	I feel more confident expressing my ideas in English conversations after using the system.	(Liu, 2013)
Negative Emotion	negative-emo-frustration	I feel frustrated when the system immediately corrects my grammar mistakes	(Ryan and Henderson, 2018)
	negative-emo-annoyance	I feel annoyed when the system immediately corrects my mistakes	(Agudo and de Dios, 2013)
Usefulness	usefulness-feedback	I think the grammar correction feedback during the chat is useful.	(Agudo and de Dios, 2013)
	usefulness-suggestion	I get useful suggestions about how to improve my grammar in English conversations	
Enjoyment	enjoyment-1	I enjoyed talking with the chatbot.	(Saadé et al., 2008)
	enjoyment-2	Talking with the chatbot was pleasant.	
Intention to use	intention-to-use-1	I would like to use this system again.	(Rosenthal and Ratan, 2022)
	intention-to-use-2	I am not interested in using this system again.	

## G GEC model performance

Table 10: Performance of our T5 GEC model by grammar error type following ERRANT’s error code.

Type	TP	FP	FN	Prec	Rec	F <sub>0.5</sub>
M:ADJ	0	1	0	0	1	0
M:ADV	1	0	6	1	0.14	0.45
M:CONJ	0	0	5	1	0	0
M:CONTR	0	0	6	1	0	0
M:DET	7	13	37	0.35	0.16	0.28
M:NOUN	0	2	1	0	0	0
M:NOUN:POSS	0	0	3	1	0	0
M:OTHER	1	1	25	0.5	0.04	0.15
M:PART	0	0	1	1	0	0
M:PREP	6	2	16	0.75	0.27	0.56
M:PRON	0	7	22	0	0	0
M:VERB	2	5	14	0.29	0.13	0.23
M:VERB:FORM	5	7	8	0.42	0.38	0.41
M:VERB:TENSE	1	1	4	0.5	0.2	0.38
R:ADJ	1	2	10	0.33	0.09	0.22
R:ADJ:FORM	1	1	4	0.5	0.2	0.38
R:ADV	3	0	9	1	0.25	0.63
R:CONJ	0	0	1	1	0	0
R:DET	9	0	17	1	0.35	0.73
R:MORPH	8	1	29	0.89	0.22	0.55
R:NOUN	2	4	31	0.33	0.06	0.18
R:NOUN:INFL	2	1	3	0.67	0.4	0.59
R:NOUN:NUM	17	16	32	0.52	0.35	0.47
R:NOUN:POSS	0	0	1	1	0	0
R:OTHER	3	15	119	0.17	0.02	0.08
R:PART	0	0	6	1	0	0
R:PREP	26	10	45	0.72	0.37	0.60
R:PRON	0	0	15	1	0	0
R:SPELL	55	26	120	0.68	0.31	0.55
R:VERB	3	3	29	0.5	0.09	0.27
R:VERB:FORM	29	12	24	0.71	0.55	0.67
R:VERB:INFL	1	0	1	1	0.5	0.83
R:VERB:SVA	18	3	6	0.86	0.75	0.83
R:VERB:TENSE	5	3	36	0.63	0.12	0.34
R:WO	0	1	15	0	0	0
U:ADJ	0	0	1	1	0	0
U:ADV	2	2	3	0.5	0.4	0.48
U:DET	2	4	15	0.33	0.12	0.24
U:NOUN	1	2	9	0.33	0.1	0.22
U:OTHER	1	19	8	0.05	0.11	0.06
U:PART	0	0	1	1	0	0
U:PREP	4	5	6	0.44	0.4	0.43
U:PRON	0	0	5	1	0	0
U:SPACE	0	0	15	1	0	0
U:VERB	2	4	7	0.33	0.22	0.30
U:VERB:FORM	0	0	2	1	0	0
U:VERB:TENSE	1	0	1	1	0.5	0.83

# Enhancing Video-based Learning Using Knowledge Tracing: Personalizing Students' Learning Experience with ORBITS

Shady Shehata, David Santandreu, Philip Purnell, Mark Thompson  
Mohamed bin Zayed University of Artificial Intelligence (MBZUAI)  
Abu Dhabi, United Arab Emirates

{shady.shehata, david.santandreu, philip.purnell, mark.thompson}@mbzuai.ac.ae

## Abstract

As the world regains its footing following the COVID-19 pandemic, academia is striving to consolidate the gains made in students' education experience. New technologies such as video-based learning have shown some early improvement in student learning and engagement. In this paper, we present ORBITS predictive engine at YOURIKA company, a video-based student support platform powered by knowledge tracing. In an exploratory case study of one master's level Speech Processing course at the Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI) in Abu Dhabi, half the students used the system while the other half did not. Student qualitative feedback was universally positive and compared the system favorably against current available methods. These findings support the use of artificial intelligence techniques to improve the student learning experience.

## 1 Introduction

Looking ahead to the post-pandemic tertiary education landscape, higher education institutions ought to innovate by incorporating effective technology to maximize a more personalized video-based learning experience. One of "the most recurring challenges towards online learning" (Munoz et al., 2021, p.2) is disengagement and absence of participation. A poll of 350 university students taking synchronous (fully online) Zoom classes indicated that approximately 94% had moderate to considerable difficulty with online learning (Peper et al., 2021). Recently,

investigators have examined the effects of Video-based Learning (VBL) on student learning and engagement (Ou et al., 2019; Poquet et al., 2018). Ou et al. (2019) and Sablić et al. (2021) have for instance established that it can have a positive impact on learning, and "students' perceptions of video effectiveness significantly predicted how they perceived the overall effectiveness of the course" (Ou et al., 2019, p. 99). Studies by Tripodi (2018) on first year osteopathic students in Australia and Lacey and Wall (2021) on three B.Sc. undergraduate student groups (microbiology) in Ireland have also indicated that VBL stimulated interest and improved performance, motivation, and engagement, as it increased exam confidence and decreased exam anxiety. An additional benefit of VBL was "improved communication with their mentees and greater ability to demonstrate the experiments to their student groups" (Lacey and Wall, 2021, p.8). So far, however, there has been little discussion in the published literature about the use of AI-powered video-based learning support platforms at the postgraduate level to personalize learning paths and improve the achievement of intended learning outcomes.

Knowledge tracing involves creating models that track students' understanding over time, enabling accurate predictions of their future performance. Advancements in this area would allow personalized resources to be recommended based on individual needs, while identifying content that may be too easy or challenging, allowing for skipping or postponing. Machine learning solutions could potentially extend the benefits of high-quality personalized instruction to anyone worldwide, at no cost.

The knowledge tracing problem is inherently challenging due to the complexity of human learning, encompassing both the human brain and accumulated knowledge. Therefore, employing sophisticated models appears to be suitable.

There are two primary aims of this study: 1. To investigate student use of and engagement with ORBITS, a video-based learning (VBL) software powered by a predictive engine that uses knowledge tracing at a graduate-level research intensive university in Abu Dhabi, United Arab Emirates, and 2. To assess the extent to which ORBITS improved students' perception of their learning experience.

This investigation takes the form of a case study. The findings should make an important contribution to the field of knowledge tracing - powered video-based learning.

This paper is organized as follows: First, we review relevant literature pertaining to Video-based Learning (VBL); the next two sections present the methodology and the results of the research, respectively. Section four discusses the results, while section five concludes.

## 2 Background

### 2.1 Video-Based Learning (VBL)

Research on video-based learning (VBL) has seen substantial growth in the last few years as a result of the launch of a) educational platforms using video (e.g., Khan Academy, LinkedIn Learning, MOOC platforms such as Edx, Coursera, FutureLearn), b) short-video hosting services (e.g., Tik Tok, Snapchat) on multiple devices, c) new features to existing apps (e.g., Instagram Reels, YouTube Shorts), and d) the adoption of lecture-capture platforms (Panopto, Echo360, Kaltura) by tertiary institutions around the world. As 1.5 billion students across 165 countries (UNESCO, 2020) were asked to return home, academic staff was requested to move all their courses fully online and use videoconferencing platforms such as Zoom, Skype, WebEx, Blackboard Collaborate Ultra or Microsoft Teams, video consumption in the past two and a half years has increased exponentially. In fact, as the COVID threat receded ZOOM or Teams meetings are still more common than physical meetings and students still prefer to attend their classes online. Torre et al. (2022) argued that “multimedia content and video-based learning are

expected to take a central role in the post-pandemic world” (p.1). Research by Calonge et al. (2019) indicated how crucial videos (and analytics) were for student retention in a fully online course. Considering recent events, it is indeed becoming extremely difficult to ignore the importance of VBL and its impact on learning and teaching (Navarrete et al., 2023). A study by Cheristiyanto (2021) on 119 high school teachers of economics in the Indonesian context indicated for instance that VBL had had a positive impact on students' learning outcomes. Another study by Schmitz et al. (2021) on the use of a flipped classroom model and video learning in a surgical course by 58 adult students (29 in the control group) showed that students in the test group preparing through the video-based online platform reached significantly higher scores in their written exams. Additionally, results of a survey by Davey et al. (2020), sent to all higher specialist orthopedic trainees in Ireland, also indicated high levels of satisfaction and positive outcomes when it concluded that “over 90% of trainees agreed that the video-based distance learning is of the same quality or an improvement of previous utilized teaching styles” (p.2089). It is now well established from a variety of studies, that the personalization of the learning experience, with immediate and customized instruction or feedback, based on students' needs and interests, a) increases cognitive, emotional, and behavioral engagement, and b) improves motivation, performance, and learning outcomes. User preference modelling, knowledge tracing, and item-based collaborative filtering have been extensively used by e-commerce websites such as Alibaba, Amazon, or social media platforms and companies such as Netflix, Hulu, Instagram, or YouTube to predict and recommend videos. Pandey and Karypis (2019) defined knowledge tracing as “the task of modeling each student's mastery of knowledge concepts (KCs) as (s)he engages with a sequence of learning activities” (p.1). Collaborative filtering analyses the similarities between users and items selected (behavior, preference) to personalize suggestions. Recent articles by Zhang (2022), El Aouifi et al. (2021), Wang (2021), Li and Ye (2020), and Madani et al. (2019) showcased for instance a user-based collaborative filtering algorithm applied to a personalized learning platform. Finally, a review of the recent research literature on video-based learning by Navarrete et al. (2021, p.8) argued that

recent approaches to forecasting learning success mainly build upon deep learning techniques (e.g., multilayer perceptrons, gated recurrent units, RNNs, and LSTMs).

## 2.2 Intelligent Tutoring System

Intelligent tutoring systems (ITSs) can successfully teach skills (like algebra, computer programming, or medical diagnosis) using learning-by-doing principles, track progress, and provide learners with personalized feedback and materials adapted to their level of understanding.

Given a learner's history of past interactions with an ITS, a performance model can be developed to estimate the current level of a learner's knowledge to predict future performance. Performance models have three major purposes: (1) enabling adaptive behavior of the instructional policy, (2) displaying the learner's estimated knowledge as a means of learning support and (3) generating interpretable and actionable intelligence. Most adaptive instructional policies used in practice today rely on an estimate of a learner's performance. They either require a learner to become proficient in one topic before allowing him/her to proceed to the next one, or sequence items based on some notion of optimal difficulty. The performance model also provides interpretable and actionable insights to learning designers, educators, and educational researchers to develop the ITS further. However, building an ITS is often very time-consuming (Weitekamp et al., 2020). Researchers started exploring and implementing different methods for quantizing performance prediction for learning including Bayesian Knowledge Tracing (BKT) or Deep Knowledge Tracing (DKT) (Piech et al. 2015).

BKT is considered the baseline approach for knowledge tracing. Other research indicated that DKT also showed a 25% gain in prediction accuracy, whereas classical statistical models could match the accuracy under a constrained environment. As such, there was no standard method of predicting the learning performance of a learner. Knowledge tracing using self-attention (Wang et al., 2022) identifies the key concepts from the student's past activities that are relevant to the given Knowledge Concept (KC) and predicts his/her mastery based on the relatively few KCs that it picked (Pandey et al., 2019). Since predictions are made based on relatively few past activities, it handles the data sparsity problem

better than the methods based on RNN. For identifying the relevance between the KCs.

Given the ability of recurrent neural networks (RNNs) to use information from an input in a prediction at a much later point in time, we hypothesized that RNN models, particularly the Long Short-Term Memory (LSTM) network, could provide significant improvements in prediction accuracy regardless of the conditions and topic considered for knowledge tracing.

Further, the accuracy of prediction and stability of parameters may impact the usability of a learning performance model. For ITS to provide actionable insights, the stability of parameters has more importance than the accuracy of prediction, whereas the accuracy of prediction impacts the performance of the adaptive behavior of ITS.

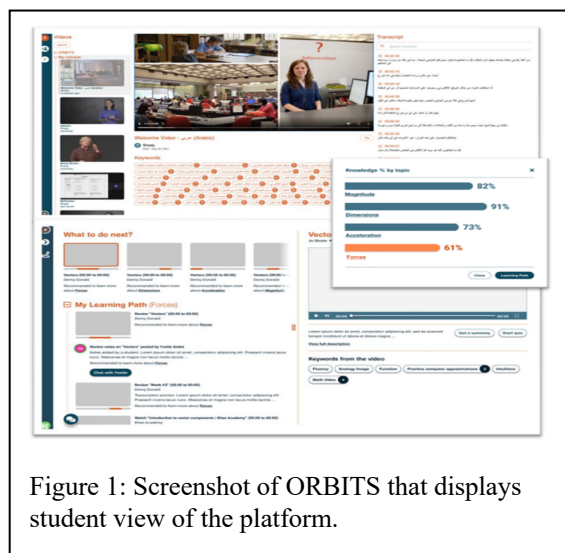


Figure 1: Screenshot of ORBITS that displays student view of the platform.

We found that the existing methods fail to overcome the lack of performance, due to a larger data size: The unsolved vanishing gradients problem hampers learning of long data sequences. The gradients carry information used in the RNN parameter update. When the gradient increasingly becomes smaller, the parameter updates become insignificant, which indicates that no significant learning is happening. The existing DKT models fail to reconstruct the observed input. As a result, even when a student performs well on an assessment, the prediction of that assessment mastery level decreases instead, and vice versa. The predicted performance for assessments across time steps is not consistent. This is undesirable and unreasonable because a student's performance is expected to transition gradually over time.

### 3 Methodology

This work proposes a personalized learning environment named ORBITS as shown in figure 1 that gathers and traces the knowledge state of the learner. We built a new model that is beyond the standard deep learning-based model based on Long Short-Term Memory (LSTM) with a new network architecture and a combination of methodologies to solve the challenges in question.

Beyond the standard approach mentioned above, we built the following four methods:

**1. Question to topic mapping:** the standard approach is to feed the data at the question level which will result in millions of combinations of feature space dimensionality. Since the number of topics is less in order of magnitude than the number of questions, in our model, we built an encoding layer of topics rather than questions to decrease the dimensionality of unique questions. The mapping between questions and topics is achieved before the training, so, at inferencing time when the question is predicted, the model will measure the knowledge state of its topics and all the dependent topics accordingly.

**2. Representative subset of the feature space:** This improves our model performance. However, the feature space still needed to be reduced. For such a large feature space, the standard approach of the one-hot encoding became impractically large. Therefore, the existing DKT models only work on specific subjects for hundreds of topics and as mentioned earlier, are unable to scale to thousands of topics. Thus, we reconstructed the input from a series of new topic-based sampling measurements to decrease the high dimensionality of the feature space. We achieved it by sampling low-dimensional representations of a one-hot high-dimensional vector. Sampling was done by picking topics that are dependent on other topics, as explained in the next step #3, our topic-based encoding layer. This means that based on fewer answers to questions/topics, the answer will be predicted, which will decrease the sparsity significantly.

**3. Topic-based encoding:** Takes two topics/questions answers, turns them into a matrix where the answers of one topic/question form the columns, and the answers of another topic/question form the rows to understand how this topic relates to others. This improves the model predictions as it can identify all the dependencies among the

knowledge states of the topics and can measure these dependencies inherently in the model.

**4. Student knowledge context:** We hypothesized that the standard approach lacks accuracy in high dimensionality since it does not take the learning context into account. And since the objective of the model is to predict what the student needs to learn on the next topic knowledge state, context is key. We, therefore, go beyond the standard approach to capture the context of the student knowledge state. Since students tend to forget topics, in what is often referred to as cognitive load (Hultberg et al., 2008), we want to preserve the knowledge state context of the topics that are answered to emphasize the recent knowledge states that have been answered. We went beyond the standard approach and ordered the input to relate a student's future interaction with topic/question to their past interaction. In this case, the model creates a representation that learns about the learning context across the topics from historic responses. The model architecture is shown in figure 2.

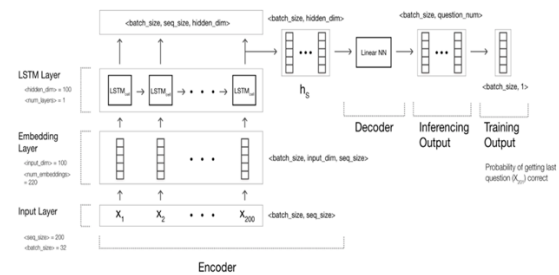


Figure 2: Model Architecture

#### 3.1 Scaling

We were looking to scale multiple thousands of topics and still achieve an AUC that is not less than ~0.7. Personalization needs inspired researchers to propose AI models to understand the knowledge state which is the “Knowledge Tracing” area. Recently, the spotlight has shone on comparisons between traditional, interpretable models such as Bayesian Knowledge Tracing (BKT) and its variants, and complex, opaque neural network models such as Deep Knowledge Tracing (DKT).

#### 3.2 Research Design

The research used in this article is an exploratory qualitative case study (Yin, 2018). This case study was conducted at a graduate-level research intensive university in the United Arab Emirates.

The purpose of this case study was to explore students' adoption of ORBITS.

The research questions being examined in this study were:

- How do students engage with the ORBITS platform?
- How do students perceive their learning experience, as affected by ORBITS predictive engine?

This study focused on contemporary events as seen through the eyes of the participants (students using ORBITS), supplemented by engagement data from ORBITS and a student feedback survey.

The case study presented here is of an exploratory nature and uses an embedded, single-case design. This design allows for the exploration of several units of analysis within the case. Specifically, there are two embedded units of analysis: 1. Student engagement with ORBITS, and 2. Student perception of the learning experience.

### 3.3 Context of the Study and Participants

This study involved postgraduate students in a course on speech processing within the MSc in Natural Language Processing (NLP).

The course is compulsory for postgraduate students in a variety of disciplines at Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI) in Abu Dhabi. Participants were informed of the study's purpose, the procedures involved, and were asked to sign a consent form. Pseudonyms (Participant - P 1, Participant - P 2, etc.) were used throughout the study to protect the participants' anonymity.

## 4 Results

This section illustrates the different methods that are used to improve the results.

**1. Question to topic mapping:** After analyzing the results, we found that the feature space still needs to be reduced. For such a large feature space, the standard approach of the one-hot encoding became impractically large, and the predictions are not consistent. This led us to find a better approach to decrease the feature space.

**2. Representative subset of the feature space:** We decreased the high dimensionality of the feature space by reconstructing the input from a series of new topic-based sampling measurements. After analyzing the results, we found that the sampling approach misses important topics that are

prerequisites to the next predicted topic in the training phase. This led us to find a better approach to selecting the prerequisite topics.

**3. Topic-based encoding:** After analyzing the results, we found that prediction performance starts with high accuracy but decreases over time. We made a further analysis to capture this behavior across students or per 1 student. We were able to segregate one student's results and found the prediction performance improves. This led us to find a better approach to address the student context issue.

**4. Student knowledge context:** After analyzing the results, we did several experiments on another four datasets as shown in Table 1 below to fine-tune the model hyper-parameters and make sure the model performance is solid. This led us to find a better approach for training the model on a large scale.

**5. Machine learning operations (ML Ops) pipeline:** Testing model against real-world datasets

We then implemented an end-to-end machine learning operations pipeline to facilitate large-scale training of hundreds of our models and modification of the model hyperparameters. The end-to-end pipeline was configured to handle the different steps to build the Knowledge Tracing system from pre-processing raw data, training a model, and deploying the system on a cloud platform.

We built a benchmarking tool to compare the different techniques with the baseline approach, as shown in Table 1 below.

	# of Topics /Skills	BKT (baseline) AUC (%)	DKT AUC (%)	Self-Attention AUC (%)
ASSIST2009	124	0.630	81.81 ± 0.10	84.20 ± 0.10
<a href="https://sites.google.com/site/assistmentsdata/home/2009-2010-assistment-data?authuser=0">https://sites.google.com/site/assistmentsdata/home/2009-2010-assistment-data?authuser=0</a>				
ASSIST2015	100	0.630	72.94 ± 0.05	82.09 ± 0.03
<a href="https://sites.google.com/site/assistmentsdata/datasets/2015-assistments-skill-builder-data">https://sites.google.com/site/assistmentsdata/datasets/2015-assistments-skill-builder-data</a>				
ASSISTmentsChall	102	0.640	72.29 ± 0.06	75.70 ± 0.32
<a href="https://sites.google.com/view/assistmentsdata/missing">https://sites.google.com/view/assistmentsdata/missing</a>				



STATICS	1223	0.540	80.87 ± 0.30	84.50 ± 0.31
<a href="https://pslcdatashop.web.cmu.edu/Project?id=48">https://pslcdatashop.web.cmu.edu/Project?id=48</a>				

Table 1: Comparison of the different techniques to improve the KT results.

We also surveyed students to gather feedback about their experience using ORBITS. Table 2 shows their comments.

<b>P1:</b>	“From my experience using ORBITS, it makes my revision and search become much easier. As for my feedback, this portal has been performing well and better than our current Moodle, and I can't wait for the future features to be implemented. I am not sure if there's any regulation regarding the availability and accessibility of the classes, but it would be nice if we students could access all lectures from any other courses that we did not take too”.
<b>P2:</b>	“It is a fantastic platform for reviewing video courses. Transcript searching and speeding up is very useful practically and can save us a lot of time. Plus, the recognition accuracy is high”.
<b>P3:</b>	“The platform looks quite stylish, and it is very convenient to search for the necessary information in the video. Also, the video download speed is very good. One of the most important advantages is the search for the necessary information by using keywords, which are also displayed at the bottom of the video. This pleased and surprised me. The speed of finding keywords is fantastic which makes the learning easier and will save a huge amount of time. The bias of this platform is made for learning regardless of the specialty and type of activity. I would probably use it only for learning since I don't see any other alternative directions for applications now”.
<b>P4:</b>	“The system is impressive and based on cutting-edge technology. Especially, the customized report on understanding and knowledge of a specific topic is amazing”.
<b>P5:</b>	“I look forward to using it more once the second semester starts and we get to have live lectures again. While using the offline videos feature, I liked that the platform is very responsive and not laggy when navigating the videos using the transcription generated by ORBITS”.

Table 2: Qualitative student feedback

Student feedback, albeit from a small sample, was overwhelmingly positive. Students spoke of the advantages the ORBITS platform provided them in their acquisition of course content (e.g., transcriptions), especially the speed (accuracy and responsiveness) with which they were able to locate specific content for learning and revision purposes. One student noted that “the customized report on understanding and knowledge of a specific topic is amazing”. Students also noted that the user interface aided efficient and strategic use of their time.

## 5 Discussion

Defining the input sequence and the way it is architected in the LSTM is the key to defining how the LSTM is used. Hence, we used LSTM in a different manner by defining the sequence in two ways: horizontally across topics and vertically across learning time per topic.

The topic-based encoding provides a sequence of topics based on their dependencies as prerequisites. Student knowledge context provides the chronological order of the student learning sequence in each topic over time and hence, across topics. This unique sequence definition enabled us to go beyond the standard approach by using LSTM in a unique manner in knowledge tracing. These two sequences would not have been possible without (1) the Question to topic mapping that enabled us to work at the level of the topic rather than questions, and (2) the Representative subset of the feature space that enabled the capability of selecting prerequisite topics.

## 6 Conclusion

The aim of the present research was to examine student use of and engagement with ORBITS and the extent to which ORBITS predictive engine improved students' perception of their learning experience. Positive initial feedback from participants indicated that ORBITS's ease of use, responsiveness, and usefulness were the three main factors of learning satisfaction. In fact, previous research has shown that learning satisfaction often correlates with performance and achievement of learning outcomes. Based on our initial exploratory findings, we can conclude that students using ORBITS are more engaged, more satisfied with their learning experience and may achieve higher assessed learning outcomes than students not using ORBITS.

## Limitations

This study reports early results and was based on a limited cohort of students from one master's course. The generalizability of these results is therefore subject to certain limitations. Follow on studies will test the system with larger samples and different disciplines to add weight to any significance of the results. Notwithstanding the relatively limited sample, this work offers valuable insights into how a video-based student support

platform that uses knowledge tracing improved graduate students' perception of their learning experience.

As the students finish their course, we will collect additional quantitative data in the form of final student grades. These will be compared between students who used the system and those who did not. We will also compare the final grades of students who used the system between the course in which they used it and other courses in which they did not. It also remains unclear what influence the course content has on the students' experience. To examine this element further, the system should be tested on multiple courses imparted by different lecturers, and in varying subject fields.

There may be an inherent positive opinion of AI-powered technologies by students at an AI university. To test that hypothesis, the system should be provided to students at other universities in subject areas not related to AI. Several questions, however, remain to be answered. Further research should be undertaken to test several hypotheses, for instance, whether Perceived Ease of use (PEoU) and Perceived usefulness (PU) would predict the Attitude towards Usage (AtU) of ORBITS.

## References

- Calonge, D. S., Riggs, K. M., Shah, M. A., & Cavanagh, T. A. (2019). Using Learning Analytics to Improve Engagement, Learning, and Design of Massive Open Online Courses. In *Fostering Multiple Levels of Engagement in Higher Education Environments* (pp. 76-107). IGI Global.
- Cheristiyanto, C. (2021). The Effectiveness of Video-Based Learning Media to Increase Student Economic Learning Outcomes During the Covid-19 Pandemic. *Economic Education Analysis Journal*, 10(3), 394-403. <https://doi.org/10.15294/eeaj.v10i3.47899>
- Davey, M. S., Cassidy, J. T., Lyons, R. F., Cleary, M. S., & Mac Niocaill, R. F. (2020). Changes to training practices during a pandemic-the experience of the irish national trauma & orthopaedic training scheme. *Injury*, 51(10), 2087-2090. <https://doi.org/10.1016/j.injury.2020.07.016>
- Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, 319-340. <https://doi.org/10.2307/249008>
- El Aouifi, H., Es-Saady, Y., El Hajji, M., Mimis, M., & Douzi, H. (2021, May). Toward student classification in educational video courses using knowledge tracing. In *Business Intelligence: 6th International Conference, CBI 2021, Beni Mellal, Morocco, May 27–29, 2021, Proceedings* (pp. 73-82). Cham: Springer International Publishing. [https://doi.org/10.1007/978-3-030-76508-8\\_6](https://doi.org/10.1007/978-3-030-76508-8_6)
- Hultberg, P., Calonge, D. S., & Lee, A. E. S. (2018). Promoting long-lasting learning through instructional design. *Journal of the Scholarship of Teaching and Learning*, 18(3). <https://doi.org/10.14434/josotl.v18i3.23179>
- Lacey, K., & Wall, J. G. (2021). Video-based learning to enhance teaching of practical microbiology. *FEMS Microbiology Letters*, 368(2), fnaa203. <https://doi.org/10.1093/femsle/fnaa203>
- Li, J., & Ye, Z. (2020). Course recommendations in online education based on collaborative filtering recommendation algorithm. *Complexity*, 2020. <https://doi.org/10.1155/2020/6619249>
- Madani, Y., Erritali, M., Bengourram, J., & Sailhan, F. (2019). Social collaborative filtering approach for recommending courses in an E-learning platform. *Procedia Computer Science*, 151, 1164-1169. <https://doi.org/10.1016/j.procs.2019.04.166>
- Munoz, K. E., Wang, M. J., & Tham, A. (2021). Enhancing online learning environments using social presence: evidence from hospitality online courses during COVID-19. *Journal of Teaching in Travel & Tourism*, 21(4), 339-357. <https://doi.org/10.1080/15313220.2021.1908871>
- Navarrete, E., Nehring, A., Schanze, S., Ewerth, R., & Hoppe, A. (2023). A Closer Look into Recent Video-based Learning Research: A Comprehensive Review of Video Characteristics, Tools, Technologies, and Learning Effectiveness. *arXiv preprint arXiv:2301.13617*. <https://doi.org/10.48550/arXiv.2301.13617>
- Navarrete, E., Hoppe, A., & Ewerth, R. (2021). A review on recent advances in video-based learning research: Video features, interaction, tools, and technologies. In *CIKM 2021 Workshops co-located with 30th ACM International Conference on Information and Knowledge Management (CIKM 2021)*, November 1-5, 2021, Gold Coast, Queensland, Australia (Vol. 3052, p. 7). Aachen, Germany: RWTH Aachen. <http://dx.doi.org/10.34657/9171>
- Ou, C., Joyner, D. A., & Goel, A. K. (2019). Designing and Developing Video Lessons for Online Learning: A Seven-Principle Model. *Online Learning*, 23(2), 82-104.
- Pandey, S., & Karypis, G. (2019). A self-attentive model for knowledge tracing. *arXiv preprint*

- arXiv:1907.06837.  
<https://doi.org/10.48550/arXiv.1907.06837>
- Peper, E., Wilson, V., Martin, M., Rosegard, E., & Harvey, R. (2021). Avoid Zoom fatigue, be present and learn. *NeuroRegulation*, 8(1), 47-47. <https://doi.org/10.15540/nr.8.1.47>
- Piech, C., Spencer, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. & Sohl-Dickstein, J (May 2021). Deep knowledge tracing. arXiv preprint arXiv:1506.05908 <https://doi.org/10.48550/arXiv.1506.05908>
- Poquet, O., Lim, L., Mirriahi, N., & Dawson, S. (2018, March). Video and learning: a systematic review (2007-2017). In Proceedings of the 8th international conference on learning analytics and knowledge (pp. 151-160). <https://doi.org/10.1145/3170358.3170376>
- Ragin, C. C. (1992). Introduction: Cases of "What is a case?". In C. C. Ragin & H. S. Becker (Eds.), *What is a Case: Exploring the Foundations of Social Inquiry* (pp. 1-17). New York, NY: Cambridge University Press.
- Sablić, M., Miroslavljević, A., & Škugor, A. (2021). Video-based learning (VBL)—past, present and future: An overview of the research published from 2008 to 2019. *Technology, Knowledge, and Learning*, 26(4), 1061-107. <https://doi.org/10.1007/s10758-020-09455-5>
- Schmitz, S. M., Schipper, S., Lemos, M., Alizai, P. H., Kokott, E., Brozat, J. F., Neumann, U.P; & Ulmer, T. F. (2021). Development of a tailor - made surgical online learning platform, ensuring surgical education in times of the COVID19 pandemic. *BMC surgery*, 21(1), 1-6. <https://doi.org/10.1186/s12893-021-01203-5>
- Torre, I., Galluccio, I., & Coccoli, M. (2022, June). Video augmentation to support video-based learning. In Proceedings of the 2022 International Conference on Advanced Visual Interfaces (pp. 1-5). <https://doi.org/10.1145/3531073.3531179>
- Tripodi, N. (2018). First-year osteopathic students' use and perceptions of complementary video-based learning. *International Journal of Osteopathic Medicine*, 30, 35-43. <https://doi.org/10.1016/j.ijosm.2018.09.004>
- UNESCO. (2020, March 26). UNESCO rallies international organizations, civil society, and private sector partners in a broad Coalition to ensure #LearningNeverStops. <https://en.unesco.org/news/unesco-rallies-international-organizations-civil-society-and-private-sector-partners-broad>
- Wang, H. (2021, September). Design and implementation of web online education platform based on user collaborative filtering algorithm. In 2021 4th International Conference on Information Systems and Computer Aided Education (pp. 2911-2918). <https://doi.org/10.1145/3482632.3487539>
- Weitekamp, D., Harpstead, E., & Koedinger, K. R. (2020, April). An interaction design for machine teaching to develop AI tutors. In Proceedings of the 2020 CHI conference on human factors in computing systems (pp. 1-11). <https://doi.org/10.1145/3313831.3376226>
- X. Wang, Z. Zetao, Z. Jia & Y. Weihao (2023). What is wrong with deep knowledge tracing? Attention-based knowledge tracing. *Appl. Intell.* <https://doi.org/10.1007/s10489-022-03621-1>
- Yin, R. K. (2018). *Case study research and applications*. Sage.
- Zhang, Q. (2022). Construction of Personalized Learning Platform Based on Collaborative Filtering Algorithm. *Wireless Communications and Mobile Computing*, 2022. <https://doi.org/10.1155/2022/5878344>

# Enhancing Human Summaries for Question-Answer Generation in Education

Hannah Gonzalez, Liam Dugan, Eleni Miltsakaki, Zhiqi Cui,  
Jiaxuan Ren, Bryan Li, Shriyash Upadhyay, Etan Ginsberg, Chris Callison-Burch  
University of Pennsylvania

`hannahgl, ldugan, elenimi, zhiqicui, rjx, bryanli, shriyash, etangins, ccb@seas.upenn.edu`

## Abstract

We address the problem of generating high-quality question-answer pairs for educational materials. Previous work on this problem showed that using summaries as input improves the quality of question generation (QG) over original textbook text and that human-written summaries result in higher quality QG than automatic summaries. In this paper, a) we show that advances in Large Language Models (LLMs) are not yet sufficient to generate quality summaries for QG and b) we introduce a new methodology for rewriting bullet point student notes into fully-fledged summaries and find that our methodology yields higher quality QG. We conducted a large-scale human annotation study of generated question-answer pairs for the evaluation of our methodology. In order to aid in future research, we release a novel [dataset](#) of 9.2K human annotations of generated questions.

## 1 Introduction

Automated generation of question-answer pairs for education can be used to assist students with self-guided reviews of educational materials or to support instructors with the creation of assessment materials. A key challenge for these question generation (QG) models is to ensure the relevancy of generated questions. Most human evaluation of QG models often emphasizes the grammaticality and fluency of the generated questions, rather than their relevance (Subramanian et al., 2017). For educational applications, this shortcoming is critical.

A recent study by Dugan et al. (2022) showed that providing QG models with human-written summaries as input, instead of original textbook text, increases question relevance, acceptability, and interpretability. The study also demonstrated that using automatically generated summaries as input improved QG quality over original textbook input, but not as much as human-written summaries.

We investigate whether advances to large language models (LLMs) like GPT-3 have closed this gap and introduce a novel methodology for generating summaries using student notes in the form of bullet points as input.

The main contributions of our research are:

1. We find that using human summaries as input to QG models still results in higher quality questions than generated summaries, even when using GPT-3 for summarization.
2. We propose a new methodology, Bull2Sum, that rewrites bullet point student notes into fully-fledged summaries.
3. We show that our Bull2Sum method of generating summaries as input to QG results in high-quality question-answer pairs.
4. We conduct a large-scale human evaluation study of generated question-answer pairs using our method and baselines.
5. To assist in future research, we release two [datasets](#): a dataset with 9.2K human annotations of generated questions, as well as a dataset with summaries written by 392 students for 96 sub-chapters of two textbooks.

## 2 Related Work

Prior work in question generation has focused primarily on using sequence-to-sequence models to generate questions from a given context passage. These methods can either be answer-aware (i.e., an answer span is given to the model, along with the passage) or answer-agnostic (i.e., just the context passage is given). Our work focuses on the latter case, in which the model has the much more challenging task of generating the answer as well as the question.

Subramanian et al. (2018) accomplished this by decomposing the generation process into two stages: answer-phrase extraction and answer-aware

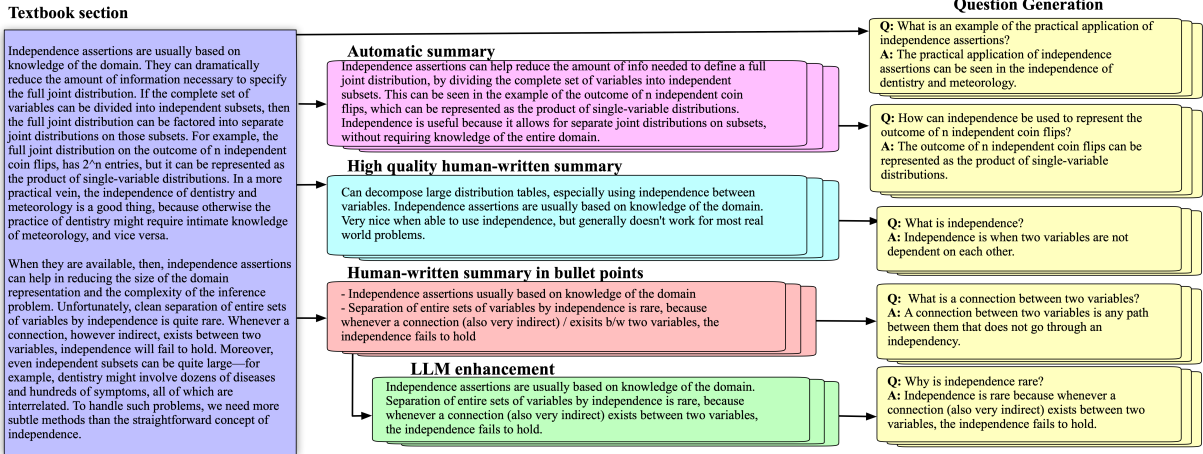


Figure 1: Different types of summaries such as automatic summaries, fully-fledged human-written summaries, human-written summaries in bullet points, LLM enhanced summaries (with our proposed method), and textbook text as used input to QG.

QG. Follow-up work from Sun et al. (2018) introduced a position-aware component to localize answers in the input context. Work by Wang et al. (2020) added joint training between the two stages of the pipeline. Other work has found that transforming the input context passage can aid in answer-phrase extraction. Lewis et al. (2021) filtered out passages that are unlikely to contain answers to human-written questions. Qu et al. (2021) generated coarse keyphrases from input passages to help guide the answer extraction model. Zhao et al. (2022) used an “event-centric” summarizer to generate a sequence of events, allowing them to ask better questions.

More recently, Dugan et al. (2022) showed that providing answer extraction models with human-written or LM-generated summaries significantly improved the relevance and interpretability of generated questions. We build on this insight and further investigate the gap in question quality between human-written summaries and LM-generated summaries. Dugan et al. used a BART model (Lewis et al., 2019) for automatic summarization. However, recent work suggests that summaries generated by large language models such as GPT-3 are overwhelmingly preferred by human annotators (Goyal et al., 2022). In what follows, we report the results of experiments that we conducted to evaluate whether Large Language Models can, indeed, generate quality summaries for the task of generating question-answer pairs for educational materials.

### Step - Description

- 1. Zero-shot** - We generated new summaries from the human-written bullet style summaries with GPT-3 using the following prompt: *“Here’s an outline, please expand it into full sentences and paragraphs: {human-written summary in bullet style with incomplete sentences}”*
- 2. Few-shot** - We reviewed 10 examples by fact-checking and removing repeated phrases. We added these examples to the prompt and then generated 100 more summaries out of the hand-written summaries in bullet style.
- 3. Fine-tuning** - We fine-tuned GPT-3’s Davinci model with the 100 summaries generated from the few-shot stage. The format of the fine-tuned model was the following: StudentSummary: <bullet-point summary> GPT3Summary: <paragraph style generated summary>

Table 1: Description of bootstrapping process to modify the human-written summary style

## 3 Methodology

As mentioned earlier, the central goal of this study is to address the problem of providing quality summaries of educational materials to QA models in order to generate important and relevant QA pairs. To investigate this problem, we ran two groups of experiments. First, we evaluated if GPT-3 generates better QA pairs than T5 which was used for the same task in prior work. In the second group of experiments, we investigated the impact of different types of input on the quality of the generation of QA pairs in addition to different ways of obtaining summaries. To this end, we collected summaries written by college students on course textbooks and classified them into two major categories: fully-fledged summaries and bullet-point summaries. Fully-fledged summaries consisted

of complete grammatical sentences that formed a coherent paragraph. Bullet-point summaries consisted of bullet points or other fragments taken as short notes. In addition to these two types of input generated by humans, we introduced and compared a new method of generating summaries from bullet-point notes, which we call Bull2Sum (from bullets to summaries). Bull2Sum takes as input bullet-point summaries and rewrites them into fully-fledged summaries.

### 3.1 Human-written Summaries

We collected human-written summaries from a total of 570 undergraduate and Master’s students enrolled in a graduate-level Artificial Intelligence course. Students wrote summaries of 56 sections of 14 chapters of the [Russell and Norvig \(2020\)](#) textbook "Artificial Intelligence: A Modern Approach" and 40 sections of 6 chapters of the [Jurafsky and Martin \(2022\)](#) textbook "Speech and Language Processing." The collected summaries varied widely in terms of style. Some students wrote fully-fledged summaries with complete sentences organized into paragraphs. Others summarized the chapters in the form of bullet-point notes. The students were incentivized to write quality summaries because they were allowed to use them as supplementary material during the final exam. We release the [summaries](#) of a total of 392 students who agreed to share their anonymized summaries with the research community.

### 3.2 Bootstrapping Training Data for LLMs

In order to generate in-domain data for fine-tuning large language models, such as GPT-3, we employed a bootstrapping approach. We first generated a small amount of data pairs by using the model in a zero-shot fashion. We then manually reviewed the generated examples by fact-checking and removing repeated phrases. We then used this filtered set of synthetic data as in-context examples to generate a larger set of high-quality few-shot data. We used this final set of examples as our fine-tuning dataset.

### 3.3 Fine-tuned Model for Rewriting Bullet Points into Summaries

We introduce a fine-tuned model, Bull2Sum, that we trained in order to rewrite summaries written in bullet points or short notes into fully-fledged summaries. We built this model by fine-tuning GPT-3 using the same bootstrapping approach described

Step	Description
1.	<b>Zero-shot</b> - We generated QA pairs with GPT-3 using the prompt "Write 5 to 10 questions along with their corresponding answers from the summary." + "Summary: " + <i>student_summary</i> + "Question: <Text of question.> + "Corresponding answer: <Text of corresponding answer.>"
2.	<b>Few-shot</b> - We reviewed 20 examples by fact-checking and formatting. We added these examples to the prompt and then generated QA pairs out of summaries generated by Bull2Sum.
3.	<b>Fine-tuning</b> - We fine-tuned a model with QA pairs generated from the few-shot stage.

Table 2: Description of bootstrapping process to generate QA pairs from a text.

in the previous section.<sup>1</sup> Table 1 outlines and compares all the methods in our experiments.

### 3.4 Question Generation Models

For question generation, we again used a bootstrapping procedure to fine-tune GPT-3 to perform answer-agnostic question generation. We outline this procedure in Table 2. We generated questions from this model and compared them to questions generated from the same fine-tuned T5 model used in [Dugan et al. \(2022\)](#).

## 4 Experiments

We compare the performance of two LLMs trained to do QG in 5 text input conditions. So, we ran a total of 10 experiments. Each condition is a different type of input to the model, including a condition with summaries generated by a new model that we fine-tuned, Bull2Sum, which rewrites bullet points or short notes into fully-fledged sentences. We describe this model in Section 3.3.

#### Text input conditions

1. Original text from textbook.
2. Zero-shot summary generated by GPT-3.
3. Fully-fledged human-written summary.
4. Bullet-point human-written summary.
5. Summary generated by Bull2Sum.

In order to evaluate and compare the performance of both T5 and GPT-3 under the 1st condition (original text from textbook), we extracted 47 sections from the [Russell and Norvig \(2020\)](#) textbook (omitting figures, tables, and equations). For the 2nd condition, we used GPT-3 to summarize the

<sup>1</sup>We ran all the reported experiments in November 2022, using text-davinci-002.

Type of Input to QG Model	T5					GPT-3				
	Acc.	Gram.	Interp.	Rel.	Corr.	Acc.	Gram.	Interp.	Rel.	Corr.
1) Original text from textbook	35%	94%	68%	69%	52%	50%	79%	71%	77%	59%
2) GPT-3 generated summary from textbook text	48%	93%	72%	76%	59%	67%	93%	84%	86%	75%
3) Fully-fledged human-written summary	44%	88%	70%	85%	58%	<b>73%</b>	95%	<b>92%</b>	<b>95%</b>	79%
4) Bullet-point human-written summary	50%	86%	72%	88%	61%	53%	93%	86%	89%	66%
5) Bull2Sum summary	55%	93%	79%	93%	67%	70%	<b>96%</b>	90%	93%	<b>80%</b>

Table 3: Evaluation of questions generation by T5 and by GPT-3 using different types of summaries as input. Humans evaluated whether the questions were Acceptable, Grammatical, Interpretable, Relevant, and Correct.

passages from the first condition with the following prompt: "Please summarize the following text using complete sentences:" For the 3rd and 4th conditions, we used 96 fully-fledged human-written summaries and 96 bullet-point human-written summaries from Russell and Norvig (56 sections) and Jurafsky and Martin (40 sections). For the 5th condition, we used our fine-tuned model Bull2Sum described in Section 3.3 on the 96 bullet-point human-written summaries. There is a one-to-one mapping in conditions 3, 4, and 5 as they are from the same textbook sections. Conditions 1 and 2 are from a subset of these textbook sections. Table 4 in the Appendix provides detailed information and statistics about the data.

## 5 Evaluation

We performed a human evaluation study to measure the QG performance of the models GPT-3 and T5 under our 5 different input conditions, as described in Section 4. We had a total of 66 annotators, all University students enrolled in an advanced Computer Science course titled *Artificial Intelligence*. Prior to the annotations, students signed a consent form to participate in the experiment and were rewarded with extra credit for their participation. Moreover, we had a training session with the students to review the guidelines and demo the annotation tool. We employed the evaluation guidelines defined in Dugan et al. (2022). For each generated QA pair, the annotators evaluated the following criteria:

1. Acceptable: Would you directly use this question as a flashcard?
2. Grammatical: Is this question grammatically correct?
3. Interpretable: Does this question make sense out of context?
4. Relevant: Is this question relevant?

5. Correct: Is the answer to the question correct?

Our team created a web-based tool (as illustrated in Appendix Figure 2) in order to increase the scalability and ease of annotations. We randomly selected 10 QA pairs generated from each of our 5 input conditions by both the T5 and GPT-3 models. We divided our 66 annotators into groups of 3, for a total of 22 groups. Each group would annotate the same group of questions generated by the different models for the same data. Given that we had 22 groups of annotators, we collected 3,080 question-answer (QA) pairs annotated, i.e., 220 QA pairs annotated per input condition. We computed pairwise inter-annotator agreement (IAA) analysis using Fleiss’s Multi- $\pi$  method (Artstein and Poesio, 2008) for finding the agreement for more than two coders and found IAA rates between 0.39–0.44 for our 5 evaluation criteria. We report the results in Table 8.

## 6 Results and Discussion

Table 3 shows the percentage of generated QA pairs where the annotations were "yes" for both GPT-3 and T5. Unsurprisingly, the larger LM GPT-3 demonstrated superior performance in the question generation task compared to T5. It produced a) higher quality flashcards, b) more questions that were coherent out of context, and c) more accurate answers. We found that fully-fledged summaries are better input than GPT-3 generated summaries, which are better than bullet-point human-written summaries. Our methodology of applying our rewriting model Bull2Summ for rewriting the bullet summaries into fully-fledged summaries results in a substantial increase in the quality of the QA pairs. Specifically, the acceptability score improves from 53% (bullet points) to 70% (nearly equal to the 73% of fully-fledged human-written summaries).

Our experiments show that although GPT-3 performs better than T5 in QG, it is not sufficient to improve a) the quality of QA pairs and b) the quality of the automated summaries as input. Carefully written human summaries are still better than automated summaries generated by GPT-3. However, our novel method of rewriting short bullet-point notes into summaries can be effectively used to generate quality QA pairs.

## 7 Limitations

In this work, we explored question generation for computer science textbooks. We have not yet explored a broader range of course subjects, and it may be that the prevalence of computer science knowledge on the Internet, including through forums like Stack Exchange, makes QG easier for this discipline than for others. Furthermore, we examine a relatively narrow range of question types. Other questions –like multiple choice questions, or compare and contrast questions– will require deeper exploration and substantial adaptation of the methodology that we proposed.

## 8 Ethics Statement

**Potential risks** : As with all large language models, the models used in our research have the potential to generate factually incorrect information. This is a potential risk given that our intended application is for education. As reported in our paper, our best-performing models produce acceptable quality flashcard questions only 70% of the time. The remaining 30% is significant enough that manual review by course is necessary before questions are deployed to students.

**Intended Use** : Our models and methods shown here are for research purposes only. They should not be deployed in the real world as solutions without further evaluation.

**Potential applications** : Bull2Sum could be utilized in the field of education to convert course slides into summaries, which can then be used to generate pertinent and significant questions for the course. This application could enhance and facilitate students’ exam preparation.

## Acknowledgements

This research is based upon work supported in part by the DARPA KAIROS Program (contract FA8750-19-2-1004), the DARPA LwLL Program

(contract FA8750-19-2-0201), the IARPA HIATUS Program (contract 2022-22072200005), and the NSF (Award 1928631). Approved for Public Release, Distribution Unlimited. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA, IARPA, NSF, or the U.S. Government.

Furthermore, the University of Pennsylvania provided valuable support for this research through the Vagelos Undergraduate Research Grant.

Additionally, we would like to extend our gratitude to Suraj Patil for providing one of the fine-tuned question generation models (T5) that we used in our experiment.

Finally, we would like to thank Jack Collison for his helpful suggestions.

## References

- Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational linguistics*, 34(4):555–596.
- Liam Dugan, Eleni Miltsakaki, Shriyash Upadhyay, Etan Ginsberg, Hannah Gonzalez, DaHyeon Choi, Chuning Yuan, and Chris Callison-Burch. 2022. [A feasibility study of answer-agnostic question generation for education](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1919–1926, Dublin, Ireland. Association for Computational Linguistics.
- Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. [News summarization and evaluation in the era of gpt-3](#).
- Daniel Jurafsky and James H Martin. 2022. *Speech and language processing (3rd Edition Draft)*. Prentice Hall NJ.
- 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](#).
- Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenertorp, and Sebastian Riedel. 2021. [PAQ: 65 million probably-asked questions and what you can do with them](#). *Transactions of the Association for Computational Linguistics*, 9:1098–1115.
- Fanyi Qu, Xin Jia, and Yunfang Wu. 2021. [Asking questions like educational experts: Automatically generating question-answer pairs on real-world examination data](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2583–2593, Online and Punta Cana,



Dominican Republic. Association for Computational Linguistics.

Stuart Russell and Peter Norvig. 2020. *Artificial Intelligence : A Modern Approach*. Pearson, Boston.

Sandeep Subramanian, Tong Wang, Xingdi Yuan, Saizheng Zhang, Yoshua Bengio, and Adam Trischler. 2017. Neural models for key phrase detection and question generation. *arXiv preprint arXiv:1706.04560*.

Sandeep Subramanian, Tong Wang, Xingdi Yuan, Saizheng Zhang, Adam Trischler, and Yoshua Bengio. 2018. [Neural models for key phrase extraction and question generation](#). In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 78–88, Melbourne, Australia. Association for Computational Linguistics.

Xingwu Sun, Jing Liu, Yajuan Lyu, Wei He, Yanjun Ma, and Shi Wang. 2018. [Answer-focused and position-aware neural question generation](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3930–3939, Brussels, Belgium. Association for Computational Linguistics.

Bingning Wang, Xiaochuan Wang, Ting Tao, Qi Zhang, and Jingfang Xu. 2020. [Neural question generation with answer pivot](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9138–9145.

Zhenjie Zhao, Yufang Hou, Dakuo Wang, Mo Yu, Chengzhong Liu, and Xiaojuan Ma. 2022. [Educational question generation of children storybooks via question type distribution learning and event-centric summarization](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5073–5085, Dublin, Ireland. Association for Computational Linguistics.

Type of Input to QG Model	Average len of the text	Average sen- tence len	Average num of sentences	Num of T5 QA pairs	Num of GPT-3 QA pairs
1) Original text from textbook	2260	116	16	774	199
2) GPT-3 generated summary from textbook text	694	103	5	265	194
3) Fully-fledged human-written summary	784	74	9	834	374
4) Bullet-point human-written summary	930	378	4	399	279
5) Bull2Sum summary	687	89	6	605	433
6) Few-shot generated summary	751	108	7	609	447
7) Summary generated with our fine-tuned model	781	92	7	698	356

Table 4: Statistics of the different types of summaries as input. We report the average length of the text (in chars), the average sentence length (in chars), the average number of sentences, the number of T5 QA pairs, and the number of GPT-3 QA pairs.

Type of Input to QG Model	Summary
Original text from textbook	<p>27.1 The Limits of AI</p> <p>27.1.2 The argument from disability</p> <p>The “argument from disability” makes the claim that “a machine can never do X.” As examples of X, Turing lists the following: Be kind, resourceful, beautiful, friendly, have initiative, have a sense of humor, tell right from wrong, make mistakes, fall in love, enjoy strawberries and cream, make someone fall in love with it, learn from experience, use words properly, be the subject of its own thought, have as much diversity of behavior as man, do something really new.</p> <p>In retrospect, some of these are rather easy—we’re all familiar with computers that “make mistakes.” Computers with metareasoning capabilities (Chapter 5) can examine heir own computations, thus being the subject of their own reasoning. A century-old technology has the proven ability to “make someone fall in love with it”—the teddy bear. Computer chess expert David Levy predicts that by 2050 people will routinely fall in love with humanoid robots. As for a robot falling in love, that is a common theme in fiction,<sup>1</sup> but there has been only limited academic speculation on the subject (Kim et al., 2007). Computers have done things that are “really new,” making significant discoveries in astronomy, mathematics, chemistry, mineralogy, biology, computer science, and other fields, and creating new forms of art through style transfer (Gatys et al., 2016). Overall, programs exceed human performance in some tasks and lag behind on others. The one thing that it is clear they can’t do is be exactly human.</p>
GPT-3 generated summary from text	<p>The "argument from disability" claims that machines can never do X, where X is a characteristic of humans. Some examples of X include being kind, resourceful, beautiful, friendly, having initiative, having a sense of humor, being able to tell right from wrong, making mistakes, falling in love, enjoying strawberries and cream, making someone fall in love with it, learning from experience, using words properly, being the subject of its own thought, having as much diversity of behavior as man, and doing something really new.</p>

Table 5: Qualitative example of sub-chapter from original textbook text and its corresponding GPT-3 generated summary.

Type of Input to QG Model	Summary
Higher quality human-written summary	<p>Perceptron can easily compute AND and OR, but cannot handle XOR at all! This is because perceptron is a linear classifier, and thus decision boundary is a line. XOR can be calculated by a layered network of units. Example using two ReLU-based units. Can view hidden layer of the network as forming a representation for the input. Neural networks big advantage is they can automatically learn to form these representations well. Note: XOR requires network of units with non-linear activation functions, i.e. CANNOT do it with just perceptrons (simple linear units). This due to fact that many layers of linear units can be reduced to one layer with a certain weights function, which we know can't solve the XOR problem.</p>
Poorer quality human-written summary	<ul style="list-style-type: none"> <li>• perceptron</li> <li>• a simple neural unit</li> <li>• has binary output</li> <li>• does not have a non-linear activation function</li> <li>• XOR: not linearly separable</li> <li>• Non-linear activation functions</li> <li>• a network formed by many layers of purely linear units can always be reduce to a single layer of linear units with appropriate weights</li> </ul>
Summary expanded into full sentences and paragraphs by zero shot	<p>A perceptron is a simple neural unit that has binary output and does not have a non-linear activation function. The XOR problem is not linearly separable, meaning that a network formed by many layers of purely linear units can always be reduced to a single layer of linear units with appropriate weights. Non-linear activation functions are necessary in order to solve the XOR problem.</p>
Few-shot generated summary	<p>The perceptron is a simple neural unit that has binary output and does not have a non-linear activation function. The XOR problem is not linearly separable. Non-linear activation functions are important because a network formed by many layers of purely linear units can always be reduced to a single layer of linear units with appropriate weights.</p>
Summary generated with our fine-tuned model	<p>A perceptron is a simple neural unit that has a binary output and does not have a non-linear activation function. An example of a perceptron is a machine that recognizes numbers written in digits. The machine would have to be trained first by being shown examples of the correct outputs for each input. The XOR problem is an example of a perceptron that is not linearly separable. A perceptron that can solve the XOR problem is said to be bi-dimensional. A perceptron that can solve the XOR problem is said to be bi-dimensional. A network formed by many layers of purely linear units can always be reduced to a single layer of linear units with appropriate weights.</p>

Table 6: Qualitative example of our methodology to enhance summary quality.

Type of Input to QG Model	GPT-3 QA	T5 QA
Original text from text-book	Q: Are programs currently capable of doing all of these things? A: Programs are currently capable of doing some of these things, but not all of them.	Q: What is a century-old technology that can make someone fall in love with it? A: teddy bear
GPT-3 generated summary from text	Q: What is the argument from disability? A: The argument from disability claims that machines can never do x, where x is a characteristic of humans.	Q: What does the "argument from disability" claim machines can never do? A: X
Higher quality human-written summary	Q: How does a perceptron compute the and logical operation? A: A perceptron computes the and logical operation by adding the activation values of the multiple layers of units in the perceptron.	Q: Perceptron can easily compute what? A: AND and OR
Poorer quality human-written summary	Q: What does a perceptron not have? A: A perceptron does not have a non-linear activation function.	Q: What is a simple neural unit - has binary output - does not have a non-linear activation function? A: perceptron
Summary expanded into full sentences and paragraphs by zero shot	Q: What is the xor problem? A: The xor problem is a problem in which two inputs cannot be separated by a linear function.	Q: What problem is not linearly separable? A: XOR
Few-shot generated summary	Q: What is the xor problem not linearly separable? A: The xor problem is not linearly separable.	Q: A network formed by many layers of what can always be reduced to a single layer of linear units with appropriate weights? A: purely linear units
Summary generated with our fine-tuned model	Q: What is a perceptron? A: A perceptron is a simple neural unit that has binary output and does not have a non-linear activation function.	Q: A perceptron that can solve the XOR problem is said to be what? A: bi-dimensional

Table 7: Qualitative examples of QA pairs generated by GPT-3 and T5 from different types of summary or text as input.

Q: What is the uncertainty that agents need to handle in the real world?

A: The uncertainty that agents need to handle in the real world can be due to partial observability, nondeterminism or adversaries.

Would you directly use this question as a flash card?

- Yes
- No
- With Edits

Is this question grammatically correct?

- Yes
- No

Does this question make sense out of context?

- Yes
- No

Is this question relevant?

- Yes
- No

Is the answer to the question correct?

- Yes
- No

Figure 2: An example of annotation interface. You can find the annotation tutorial [here](#).

---

<b>IAA</b>	
Acceptable	0.39
Grammatical	0.44
Interpretable	0.42
Relevant	0.42
Correct	0.39

---

Table 8: Mean of pairwise agreement in all 22 groups

# Difficulty-Controllable Neural Question Generation for Reading Comprehension using Item Response Theory

Masaki Uto and Yuto Tomikawa and Ayaka Suzuki

The University of Electro-Communications

Tokyo, Japan

{uto, tomikawa, suzuki\_ayaka}@ai.lab.uec.ac.jp

## Abstract

Question generation (QG) for reading comprehension, a technology for automatically generating questions related to given reading passages, has been used in various applications, including in education. Recently, QG methods based on deep neural networks have succeeded in generating fluent questions that are pertinent to given reading passages. One example of how QG can be applied in education is a reading tutor that automatically offers reading comprehension questions related to various reading materials. In such an application, QG methods should provide questions with difficulty levels appropriate for each learner’s reading ability in order to improve learning efficiency. Several difficulty-controllable QG methods have been proposed for doing so. However, conventional methods focus only on generating questions and cannot generate answers to them. Furthermore, they ignore the relation between question difficulty and learner ability, making it hard to determine an appropriate difficulty for each learner. To resolve these problems, we propose a new method for generating question-answer pairs that considers their difficulty, estimated using item response theory. The proposed difficulty-controllable generation is realized by extending two pre-trained transformer models: BERT and GPT-2.

## 1 Introduction

Automatic question generation (QG) for reading comprehension is the task of automatically generating reading comprehension questions related to given reading passages. Various QG methods have been developed in the natural language processing (NLP) research field (Zhang et al., 2021). They have also been used in various educational systems, such as intelligent tutoring systems, writing support systems, and knowledge assessment systems (Ghanem et al., 2022; Kurdi et al., 2020; Le et al., 2014; Rathod et al., 2022; Zhang et al., 2021).

Early QG methods have relied on rule-based or template-based approaches, which use hand-crafted rules or templates to generate an interrogative question text from a declarative text (Zhang et al., 2021). However, preparing those QG methods for a target application is time-consuming and labor-intensive because achieving high-quality QG requires well-designed rules and templates for each application (Chen et al., 2021; Zhang et al., 2021). End-to-end QG methods based on deep neural networks have received wide attention as a means of overcoming this limitation (Chan and Fan, 2019; Du et al., 2017; Ushio et al., 2022; Yu et al., 2023; Zhang et al., 2021). Earlier neural QG methods were designed as sequence-to-sequence (seq2seq) models based on recurrent neural networks (RNNs) and attention mechanisms (Du et al., 2017), while recent methods are based on pre-trained transformer models (Gao et al., 2019; Ghanem et al., 2022; Lee and Lee, 2022; Rathod et al., 2022; Ushio et al., 2022), including BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019), GPT-2 (Generative Pre-trained Transformer 2) (Radford et al., 2019), BART (Bidirectional and Auto-Regressive Transformers) (Lewis et al., 2020), and T5 (Text-to-Text Transfer Transformer) (Raffel et al., 2022). Those methods have succeeded in generating fluent questions that are pertinent to given reading passages.

A representative application of how QG can be used for educational purposes is a reading tutor that automatically offers reading comprehension questions related to various reading materials (Kurdi et al., 2020; Le et al., 2014; Rathod et al., 2022; Zhang et al., 2021). This helps to focus learners’ attention on the reading materials and offers the opportunity to observe any misconceptions they might have (Kurdi et al., 2020), which supports the development of reading comprehension skills. To enhance such learning, it is useful to provide questions with difficulty levels appropriate for each

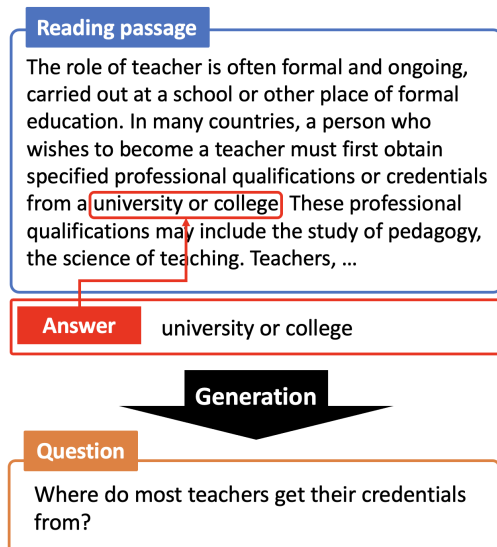


Figure 1: Conventional QG task.

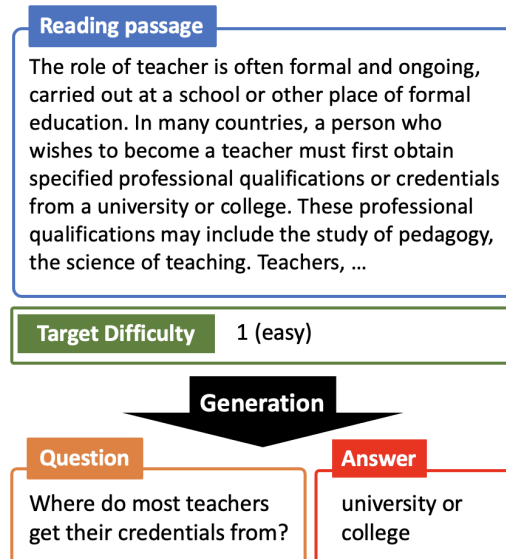


Figure 2: Our QG task.

learner’s reading ability. Such adaptivity is a core component of recent AI-based intelligent tutoring systems.

Difficulty control of QG is a relatively new task (Cheng et al., 2021; Kurdi et al., 2020), and thus previous research on difficulty-controllable QG for reading comprehension is still limited (Chen et al., 2021; Cheng et al., 2021; Gao et al., 2019). There are currently only two conventional methods; the first uses an RNN-based seq2seq model in which hidden states before its encoder are modified to receive a difficulty as input that is categorized as either easy or hard (Gao et al., 2019), and the second is a multi-hop QG (Cheng et al., 2021) that takes the question difficulty to be the number of inference steps required to answer a question and aims to generate questions while controlling the number of required inference steps. However, both methods have the following limitations that prevent them from generating questions appropriate for a learner’s ability.

1. They ignore the relation between question difficulty and learner ability, making it difficult to determine an appropriate difficulty for each learner.
2. They are answer-aware QG methods, which generate questions given a reading passage and an answer text, as illustrated in Fig. 1, and thus cannot generate question–answer pairs. Without correct answers, systems cannot score learners’ answers automatically,

meaning adaptive systems will not work efficiently. Furthermore, controlling difficulty in answer generation is also important because difficulty is a property that generally depends on both questions and answers.

To resolve these problems, we propose a new method for generating question–answer pairs that considers the difficulty associated with learners’ ability. A unique feature of our method is that it uses item response theory (IRT) (Lord, 1980), a test theory based on mathematical models, to quantify the difficulty of each question–answer pair. IRT is based on statistical models that define the relation between question difficulty and learner ability, and thus it helps us to select a difficulty appropriate for each learner’s ability. For these reasons, we aim to generate question–answer pairs while considering their difficulty, quantified by IRT. For our QG method, we first propose a method for constructing a training dataset consisting of quadruplets (reading passage, question text, answer text, and IRT-based difficulty), based on the SQuAD dataset, which is the most popular benchmark dataset for the reading comprehension QG task. Then, we propose a difficulty-controllable generation method for question–answer pairs that can be trained using this dataset. Our generation method consists of two pre-trained transformer-based models, which are extended to take IRT-based difficulty values as input: *a difficulty-controllable answer extraction model using BERT*, and *a difficulty-controllable answer-aware QG model using GPT-2*.



To our knowledge, this is the first difficulty-controllable QG method aimed at generating question–answer pairs corresponding to IRT-based difficulty.

## 2 Task Definition

The task tackled in this study is to generate a reading comprehension question and a corresponding correct answer, given a reading passage and a target difficulty value. Here, we assume that a correct answer to each question consists of a segment of text from the corresponding reading passage, as in typical answer-aware QG tasks (Rajpurkar et al., 2016). Fig. 2 shows an outline of our task.

The detailed task definition is as follows. Let a given reading passage be a word sequence  $\mathbf{r} = \{r_i \mid i \in \{1, \dots, I\}\}$ , where  $r_i$  represents the  $i$ -th word in the passage, and  $I$  is the passage text length. Similarly, let a question text  $\mathbf{q}$  and an answer text  $\mathbf{a}$  be word sequences  $\mathbf{q} = \{q_j \mid j \in \{1, \dots, J\}\}$  and  $\mathbf{a} = \{a_k \mid k \in \{1, \dots, K\}\}$ , respectively, where  $q_j$  is the  $j$ -th word in the question text,  $a_k$  is the  $k$ -th word in the answer text,  $J$  is the question text length, and  $K$  is the answer text length. Note that the answer text  $\mathbf{a}$  must be a subset of the word sequence in the reading passage  $\mathbf{r}$ , namely,  $\mathbf{a} \subset \mathbf{r}$ . Using this notation, our task is to generate a question text  $\mathbf{q}$  and an answer text  $\mathbf{a}$  given a reading passage  $\mathbf{r}$  and a target difficulty value  $b$ , where the difficulty value  $b$  is assumed to be quantified based on IRT, as explained in the introduction.

## 3 Item Response Theory

IRT (Lord, 1980) is a statistical framework used in psychometrics and educational measurement to analyze examinees’ responses to test items (*items* corresponds to *questions* in our study). One of the unique characteristics of IRT is that it estimates two types of latent factors from response data: examinee ability and item characteristics. Examinee ability refers to the latent trait or ability that the test is intended to measure, such as reading comprehension ability in our context. Item characteristics refer to the properties of test items, including their difficulty level and their ability to discriminate examinee ability. IRT uses probabilistic models, called IRT models, to estimate examinees’ abilities and item characteristics from response data that typically consist of a binary variable taking one if an examinee answers an item correctly and zero otherwise.

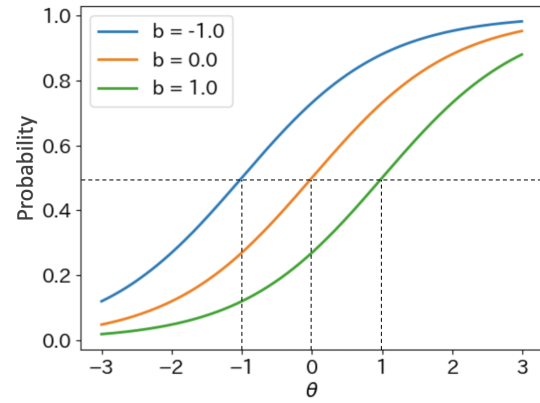


Figure 3: Item response curves for a Rasch model with different item difficulty values.

IRT has been widely used in various educational and psychological tests because it has the following typical benefits (Uto and Ueno, 2020) compared with classical test theory (a simple and traditional framework based on basic statistics such as mean, variance, and correlation coefficients): 1) IRT provides detailed information about item properties, including difficulty and discrimination, which helps test developers identify problematic items and improve test quality. 2) IRT provides accurate estimates of examinee ability and item properties. 3) The abilities of examinees who take different tests can be estimated on the same scale because examinee ability is estimated considering the effects of the items’ characteristics. 4) IRT is the basis for computerized adaptive testing (CAT), which can reduce test length and increase measurement precision by selecting appropriate items for a target examinee’s ability (van der Linden and Glas, 2010).

This study uses the Rasch model (a one-parameter logistic model), which is the most traditional and well-known IRT model. The Rasch model defines the probability that the  $m$ -th examinee correctly answers the  $n$ -th item as

$$p_{nm} = \frac{\exp(\theta_m - b_n)}{1 + \exp(\theta_m - b_n)}, \quad (1)$$

where  $b_n$  represents the difficulty of the  $n$ -th item and  $\theta_m$  represents the latent ability of the  $m$ -th examinee.

To explain the relationship between the latent ability  $\theta$  and the difficulty parameter  $b$  in the Rasch model, Fig. 3 depicts item response curves (IRCs) of the Rasch model, which are drawn by plotting the probability  $p_{nm}$ , for three different difficulty

values. In the figure, the horizontal axis shows  $\theta$ , the vertical axis shows the probability  $p_{nm}$ , and three solid curves show the IRC for three items with different difficulty values.

These IRCs show that examinees with higher  $\theta$  have a higher probability of responding correctly to each item. We can also see that the IRC shifts to the right as the item difficulty value increases, reflecting the fact that higher ability is required to correctly answer items with high  $b$ . Furthermore, under the Rasch model, the probability that an examinee with ability  $\theta$  correctly answers the question with difficulty  $b$  becomes 0.5 when  $\theta = b$ .

The IRT model parameters are generally estimated in two phases, namely, *item calibration* and *ability estimation*, in order to guarantee asymptotic consistency. Item calibration estimates the item parameters from response data by marginalizing the examinee ability  $\theta$  from the likelihood in order to ensure the asymptotic consistency of the item parameter estimates. Specifically, marginal maximum likelihood (MML) estimation using an expectation-maximization (EM) algorithm has been widely used for item calibration (Baker and Kim, 2004). Given calibrated item parameters, the ability estimation phase calculates the examinee’s ability  $\theta$ . An expected a posteriori (EAP) estimation, a type of Bayesian estimation, is generally used for the ability estimation (Fox, 2010; Uto et al., 2023).

This study aims to quantify question difficulty based on the IRT. The next section explains how to prepare the dataset with IRT-based difficulty, which is required to train our QG model.

#### 4 Creating a Dataset with IRT-based Question Difficulty

We require an appropriate dataset to construct our QG method for solving the difficulty-controllable QG task defined in Section 2. While several popular datasets have been developed for general reading comprehension QG tasks (Zhang et al., 2021), the most popular is SQuAD (Rajpurkar et al., 2016), which consists of over 100,000 question–answer pairs from Wikipedia articles. Specifically, SQuAD is a collection of triplets  $(r, q, a)$ , where each answer  $a$  is a text fragment from a corresponding reading passage  $r$  and each reading passage  $r$  corresponds to a paragraph of a Wikipedia article. However, to construct a difficulty-controllable QG method, we require a dataset consisting of quadruplets  $(r, q, a, b)$ . Thus, we first propose a method

for extending the SQuAD dataset by appending the IRT-based difficulty values for each question–answer pair. The details for doing so are as follows.

1. **Collecting response data for each question–answer pair:** We collect answers from multiple respondents to each question in the SQuAD dataset and grade those answers as correct or incorrect. Ideally, we should gather responses from a population of target learners, but this is highly expensive and time-consuming. Thus, we substitute actual learner responses with automated question–answering (QA) systems, in the same way that several previous difficulty-controllable QG studies have done (Chen et al., 2021; Gao et al., 2019).
2. **Difficulty estimation using IRT:** Using the collected response data, we estimate the question difficulty by using the Rasch model and the item calibration procedure introduced in Section 3. Note that the difficulty value generally depends on the contents of both the question and the answer.
3. **Creating a dataset with difficulty estimates:** We construct a dataset consisting of quadruplets  $(r, q, a, b)$  by appending the estimated difficulty values  $b$  into the triplets  $(r, q, a)$  of the SQuAD dataset.

## 5 Proposed Method

Our difficulty-controllable QG method, which is trained using the extended SQuAD dataset, is realized by performing the following two tasks in sequence: (1) *difficulty-controllable answer extraction* that extracts an answer text from a given reading passage while considering a target difficulty value, and (2) *difficulty-controllable answer-aware QG* that generates a question given a reading passage, an answer text, and a target difficulty value. Details of each are provided in the following sections.

### 5.1 Difficulty-Controllable Answer Extraction

We perform the difficulty-controllable answer extraction using BERT (Devlin et al., 2019). BERT is a pre-trained multilayer bidirectional transformer with 340M parameters, a transformer being a neural network architecture based on self-attention mechanisms. BERT is pre-trained on large amounts

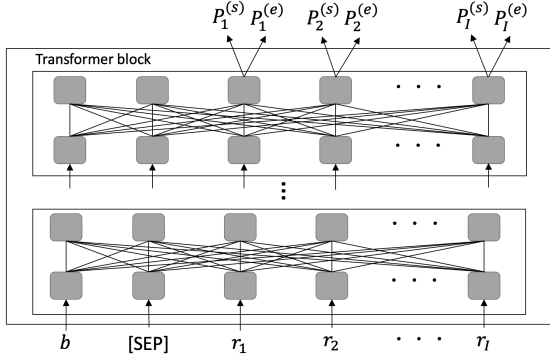


Figure 4: Difficulty-controllable answer extraction using BERT.

of text data over two unsupervised learning tasks, *masked language modeling* and *next-sentence prediction*. The pre-trained BERT can be applied to various downstream tasks by fine-tuning the model with a task-specific supervised dataset after adding task-specific output layers. We use fine-tuned BERT for the answer extraction task because BERT has been widely used before in various text extraction tasks (Srikanth et al., 2020).

To perform answer extraction using BERT, we add output layers that predict the start and end positions of the answer text within a given reading passage. Specifically, we add two dense layers with softmax activation to transform each BERT output vector, which correspond to the words within a given reading passage, into probability values for whether the word is at the start or end position of the answer text. By extracting the word sequence within the start and end positions, which take the maximum probabilities, we can extract an answer text from a given reading passage.

We control the difficulty of the answer extraction by inputting a difficulty value with the reading passage. Specifically, the input for our model is defined as

$$b, [\text{SEP}], r_1, r_2, r_3, \dots, r_I, \quad (2)$$

where [SEP] is the special token used to separate the difficulty value and the reading passage. This input is what enables the model to extract an answer text from a reading passage while considering the input difficulty value. Fig. 4 shows an outline of the answer extraction model.

We can fine-tune the answer extraction model by using a collection of triplets  $(r, a, b)$ , which can be obtained from the extended SQuAD dataset explained in Section 4. This fine-tuning is performed

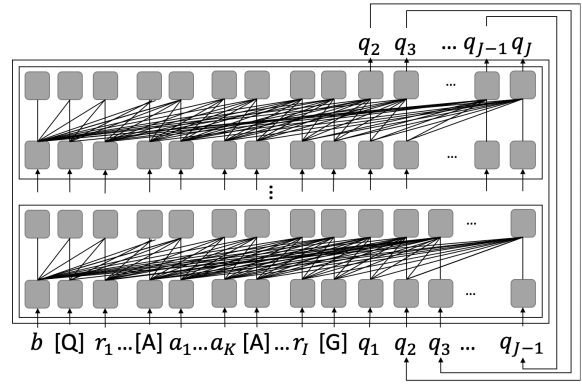


Figure 5: Difficulty-controllable answer-aware question generation using GPT-2.

by minimizing cross-entropy loss between the predicted positions of the start and end of an answer text and their true positions.

## 5.2 Difficulty-Controllable Answer-Aware Question Generation

We use GPT-2 to perform difficulty-controllable answer-aware QG. GPT-2 is a transformer-based language model with more than 1.5 billion parameters, and it is pre-trained on more than 8 million documents using an unsupervised learning process called language modeling, which sequentially predicts the next word from the current word sequence. We use GPT-2 for the QG tasks because it has been widely used before in various text generation tasks.

Conventional answer-aware QG models based on pre-trained language models (Srivastava and Goodman, 2021), including GPT-2, are implemented by designing the model’s input as

$$r_1, \dots, [A], a_1, \dots, a_K, [A], \dots, r_I, [G], \quad (3)$$

where [A] is a special token representing an answer’s start and end positions within a reading passage. [G] is also a special token representing the end of a reading passage. Conventional QG models receive this input and generate a question text after the special token [G].

To implement difficulty-control for the answer-aware QG model, we concatenate a target difficulty value to the conventional input form above using

$$b, [Q], r_1, \dots, [A], a_1, \dots, a_K, [A], \dots, r_I, [G], \quad (4)$$

where [Q] is the special token used to separate the difficulty value and the given reading passage. Given this input, the model generates a question text based on a reading passage, an answer, and a

target difficulty value. Fig. 5 presents an outline of our QG model.

We can fine-tune the answer-aware QG model by using a dataset consisting of quadruplets  $(r, q, a, b)$ , explained in Section 4. Specifically, we prepare the following format data and train GPT-2 by maximizing the log-likelihood for question texts:

$$b, [Q], r_1, \dots, [A], a_1, \dots, a_K, [A], \dots, r_I, [G], q_1, \dots, q_J. \quad (5)$$

### 5.3 Determining Appropriate Difficulty based on IRT

As explained in Section 1, IRT helps us to select a difficulty appropriate for each learner’s ability. Earlier studies on adaptive learning have demonstrated that offering questions with a difficulty at which the learner would have a 50% chance of answering correctly is the most effective approach for learning (Ueno and Miyazawa, 2018). As explained in Section 3, under the Rasch model, the probability that a learner with ability  $\theta$  correctly answers the question with difficulty  $b$  becomes 0.5 when  $\theta = b$ . Thus, we can generate questions with a difficulty appropriate for each learner using the following steps inspired by the framework of CAT (van der Linden and Glas, 2010).

1. Provide some questions randomly to a learner and collect response data.
2. Estimate the learner’s ability using the Rasch model and the response data.
3. Generate a question–answer pair by inputting the estimated ability value as the difficulty value into the proposed QG method.

Furthermore, by repeating procedures 2–3, we can enable adaptive QG.

## 6 Experiments

In this section, we demonstrate that our proposed method can generate questions and answers corresponding to target IRT-based difficulty values.

### 6.1 Data preparation

For our experiment, we first constructed an extended SQuAD dataset consisting of quadruplets  $(r, q, a, b)$  by following the procedures explained in Section 4. The original SQuAD dataset was divided into training data (90%) and test data (10%)

in advance. In this experiment, we trained QA models using the training data and constructed an extended dataset using the test data. The detailed procedures were as follows.

1. **Training QA models:** Using the SQuAD training data, we trained five different QA models: two neural models, the BERT-based model (Devlin et al., 2019) and the ALBERT-based model (Lan et al., 2020), and three feature-based models, a logistic regression model using dependency-tree features (Rajpurkar et al., 2016), a logistic regression model using selected features (Rajpurkar et al., 2016), and a sliding-window model using bag-of-words features (Richardson et al., 2013).
2. **Collecting response data for each question:** We collected answers from the five QA models for all the questions in the SQuAD test data and scored those answers.
3. **Estimating IRT-based difficulty:** Using the correct/incorrect response data, we estimated the difficulty of each question using the Rasch model. Here, we conducted the estimation using the MML method with the EM algorithm. The difficulty values were estimated to be one of six values (-3.96, -1.82, -0.26, 0.88, 2.01, 3.60), where questions with lower difficulty estimates indicate that they were easier. We linearly transformed the difficulty values estimated on the real value scale (-3.96, -1.82, -0.26, 0.88, 2.01, 3.60) to positive integer values (1, 29, 49, 64, 79, 100) to make it easier for the language models to understand the numerical inputs. Table 1 shows the ability estimates  $\hat{\theta}$  for the five QA systems, where the abilities were estimated by the EAP estimation using a Gaussian quadrature (Baker and Kim, 2004), given the calibrated item-difficulty parameters. The table shows that the abilities of the five QA systems differ greatly.

Table 1: Ability estimates  $\hat{\theta}$  of five QA systems.

	$\hat{\theta}$
BERT-based model	2.25
ALBERT-based model	1.28
Logistic regression	0.52
Logistic regression (selected features)	-0.64
Sliding-window model	-2.84

A larger variety of respondent abilities is generally effective for clearly distinguishing the difficulty among questions, suggesting that our use of these five QA systems in our experiment is reasonable. Note that ability and question difficulty are estimated assuming a standard normal distribution, meaning that these estimates distribute approximately on a scale with a mean of 0 and a standard deviation of 1.

#### 4. Creating a dataset with difficulty estimates:

We created a dataset  $\mathcal{D}$  consisting of quadruplets  $(r, q, a, b)$  by integrating the obtained IRT-based difficulty values and SQuAD test data.

## 6.2 Experimental Procedures

We conducted the following experiment using the created dataset  $\mathcal{D}$  and the original SQuAD training data.

1. Using the original SQuAD training data, we fine-tuned the proposed answer extraction model and the answer-aware QG model, ignoring the difficulty. This fine-tuning was done by removing the difficulty value from the input of the proposed models. Although this procedure is not mandatory, we applied it to improve the basic QG performance.
2. We randomly divided the dataset  $\mathcal{D}$  into parts, one 90% (designated as  $\mathcal{D}^{(train)}$ ) and the other 10% (designated as  $\mathcal{D}^{(eval)}$ ). Then, using the 90% dataset  $\mathcal{D}^{(train)}$ , we fine-tuned the difficulty-controllable answer extraction model and the difficulty-controllable answer-aware QG model, where the initial model parameters were set to the values obtained in procedure 1.
3. We generated questions and answers for each reading passage in the remaining 10% dataset  $\mathcal{D}^{(eval)}$ , given each of the six difficulty values (1, 29, 49, 64, 79, 100). Using the generated questions and answers, we conducted both an automatic evaluation and a human evaluation, which are explained below.

We used *PyTorch* and the *Transformers* library to implement the proposed models and the neural QA systems. Furthermore, we used *R* and the *TAM* package to perform the IRT parameter estimation.

Table 2: Number of questions corresponding to the six difficulty values in  $D^{(train)}$  and  $D^{(eval)}$ .

Difficulty	$D^{(train)}$	$D^{(eval)}$
1	662 (0.07)	90 (0.1)
29	2,739 (0.28)	269 (0.3)
49	1,623 (0.17)	144 (0.16)
64	2,362 (0.24)	195 (0.22)
79	1,389 (0.14)	107 (0.12)
100	909 (0.09)	81 (0.09)

Numbers in parentheses indicate ratios.

Here, we summarize the basic statistics of the datasets  $D^{(train)}$  and  $D^{(eval)}$ , which we developed in the above procedure 2 to train and evaluate our difficulty-controllable QG method. First, the number of reading passages in  $D^{(train)}$  and  $D^{(eval)}$  was 1,860 and 207, respectively. Next, the average number of questions per reading passage in  $D^{(train)}$  and  $D^{(eval)}$  was 5.21 and 4.28. Furthermore, Table 2 shows the number of questions corresponding to the six difficulty values in each dataset. From these results, we can confirm that the basic statistics and the difficulty distributions are similar between the two datasets, indicating that the dataset  $\mathcal{D}$  was randomly divided into  $D^{(train)}$  and  $D^{(eval)}$  without bias.

## 6.3 Automatic Evaluation

We performed an automatic evaluation by calculating the percentage of correct answers given by the neural QA systems (BERT-based and ALBERT-based QA models) to the questions generated for each difficulty. Fig. 6 shows the results, which indicate that the correct answer rate of QA systems

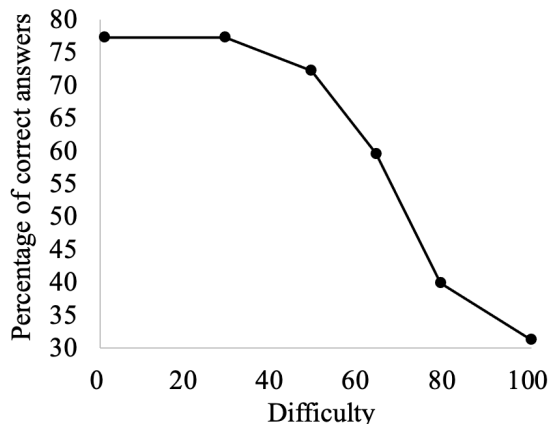


Figure 6: Percentage of correct answers by neural QA systems to questions generated for each difficulty.

Table 3: Examples of generated questions and answers for different difficulties.

Reading passage	Much of the work of the Scottish Parliament is done in committee. The role of committees is stronger in the Scottish Parliament than in other parliamentary systems, partly as a means of strengthening the role of backbenchers in their scrutiny of the government and partly to compensate for the fact that there is no revising chamber. The principal role of committees in the Scottish Parliament is to take evidence from witnesses, conduct inquiries and scrutinise legislation.
Difficulty	1 (easiest)
Question	Where is much of the work of the Scottish Parliament done?
Answer	committee
Difficulty	100 (most difficult)
Question	What is the purpose of the chairman and member of the committee?
Answer	take evidence from witnesses, conduct inquiries and scrutinise legislation

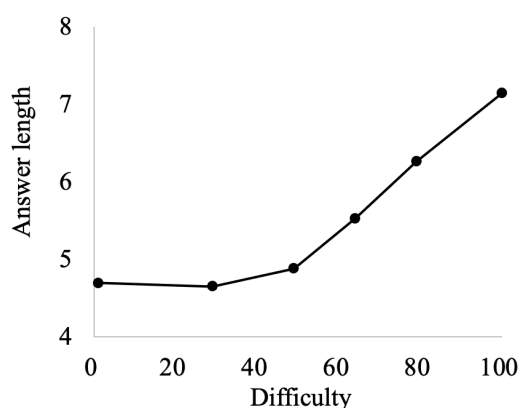


Figure 7: Average word length in generated answers for each difficulty.

decreases as the difficulty increases. This suggests that our proposed method generates questions that reflect the given difficulty.

Furthermore, we calculated the average word length in the generated answer texts for each difficulty. Fig. 7 shows the results, and these indicate that the average word length in the generated answer texts increases as the target difficulty values increase. Considering that questions with longer and more complex answers are generally difficult to correct perfectly, this result suggests that the proposed method extracts answers that reflect the specified difficulty.

Table 3 shows examples of the generated question–answer pairs when given the same reading text but different difficulty values, demonstrating that higher difficulty values correspond to longer answers.

## 6.4 Human Evaluation

For the human evaluation, we randomly selected ten reading passages from  $\mathcal{D}^{(eval)}$  and extracted question–answer pairs for the six difficulty values corresponding to each reading passage from the generated data obtained in experimental procedure 3. Then, the 60 question–answer pairs were evaluated by four human judges according to the following four evaluation metrics.

1. *Difficulty*: The subjective difficulty evaluation for each question–answer pair, graded on a scale from one to five, where smaller grades mean the question was easier.
2. *Fluency*: Evaluation of the grammatical correctness of generated questions, graded on a three-point scale: Yes, Acceptable, and No.
3. *Relevance*: Evaluation of the content relevance between generated questions and reading passages, graded on a binary scale: Yes and No.
4. *Answerability*: Evaluation of the answerability of each generated question–answer pair from a given reading passage, graded on a four-point scale: Yes, Partially, and No. Here, “Partially” indicates that the generated answer does not entirely match the correct answer for the generated question but partially includes the correct answer.

Fig 8 shows the relation between the input difficulty values and the averaged scores in the human difficulty evaluation for the generated questions. They indicate that the human subjects judged the questions generated with higher difficulty values to

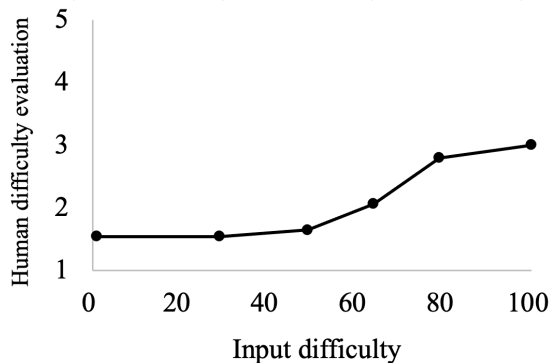


Figure 8: Human difficulty evaluation of generated question-answer pairs for each difficulty.

Table 4: The fluency, relevance, and answerability of generated questions and answers.

Fluency	Yes	Acceptable	No
	76.0%	16.3%	7.6%
Relevance	Yes	No	
	87.8%	12.2%	
Answerability	Yes	Partially	No
	67.4%	17.4%	15.3%

be more difficult. This indicates that the proposed method can appropriately control the difficulty of generated question-answer pairs.

Table 4 gives the results for *Fluency*, *Relevance*, and *Answerability*. It shows that more than 90% of the questions were generated with correct or acceptable grammar, and about 90% appropriately reflected the content of the given reading passages. Furthermore, about 70% of generated question-answer pairs were completely answerable, and about 85% were partially appropriate. These results indicate that fluency and relevance are acceptable but further improvement might be required in terms of answerability, which is planned for future work.

## 7 Conclusion

In this study, we proposed a new neural QG method that generates question-answer pairs while considering their difficulty, estimated using IRT. We also evaluated the effectiveness of this method through experiments using SQuAD.

One limitation of this study is that we used only the SQuAD dataset in our experiments. The SQuAD dataset has often been criticized because it is overly dependent on the similarity of question/answer sentences rather than on human-type reasoning, meaning it requires only superficial read-

ing skills. Thus, examining the effectiveness of our proposed method by applying it to various other datasets will be an important future task.

Furthermore, in the human evaluation experiment presented in Section 6.4, we examined only 60 question-answer pairs generated through the proposed model from ten randomly selected reading passages. The relatively small scale of the experiment is due to the high workload required for people to carefully evaluate the various properties of a large number of questions. However, in the future, we aim to conduct a larger-scale human evaluation in order to increase the reliability of the experimental results.

Although the present study used only five QA systems, the use of a larger number of QA systems with different characteristics is expected to improve the accuracy of question-difficulty estimation and provide difficulty estimates with finer granularity. Therefore, examining the effects of increasing the number and variability of QA systems will be another future direction of this research.

We also need to confirm in greater detail whether QA systems can be substituted for human learners. A comparison between IRT-based question difficulties calibrated from the responses of QA systems as well as human learners might be a plausible approach.

Another future goal is to develop a method of transforming the scale of the IRT-based difficulty, estimated based on QA systems, into a scale appropriate for a population of target learners. Such a scaling adjustment is expected to be achievable by using *equating*, which is a well-established technique in IRT.

Furthermore, our QG method is easily extended to adaptive QG systems based on the framework of computerized adaptive testing, as mentioned in Section 5.3. Developing and evaluating such an adaptive system using our QG method will also be our focus in future work.

## Acknowledgements

This work was supported by 19H05663, and 21H00898.

## References

F.B. Baker and Seock Ho Kim. 2004. *Item Response Theory: Parameter Estimation Techniques*. CRC Press, Boca Raton, FL, USA.

- Ying-Hong Chan and Yao-Chung Fan. 2019. A recurrent BERT-based model for question generation. In *Proc. Workshop on Machine Reading for Question Answering*, pages 154–162.
- Feng Chen, Jiayuan Xie, Yi Cai, Tao Wang, and Qing Li. 2021. Difficulty-controllable visual question generation. In *Proc. Web and Big Data: International Joint Conference*, pages 332–347. Springer-Verlag.
- Yi Cheng, Siyao Li, Bang Liu, Ruihui Zhao, Sujian Li, Chenghua Lin, and Yefeng Zheng. 2021. Guiding the growth: Difficulty-controllable question generation through step-by-step rewriting. In *Proc. Annual Meeting of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing*, pages 5968–5978.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186.
- Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proc. Annual Meeting of the Association for Computational Linguistics*, pages 1342–1352.
- Jean-Paul Fox. 2010. *Bayesian item response modeling: Theory and applications*. Springer, New York, NY, USA.
- Yifan Gao, Lidong Bing, Wang Chen, Michael Lyu, and Irwin King. 2019. Difficulty controllable generation of reading comprehension questions. In *Proc. International Joint Conference on Artificial Intelligence*, pages 4968–4974.
- Bilal Ghanem, Lauren Lutz Coleman, Julia Rivard Dexter, Spencer von der Ohe, and Alona Fyshe. 2022. Question generation for reading comprehension assessment by modeling how and what to ask. In *Findings of the Association for Computational Linguistics*, pages 2131–2146.
- 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*, 30:121–204.
- 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. In *Proc. International Conference on Learning Representations*.
- Nguyen-Thinh Le, Tomoko Kojiri, and Niels Pinkwart. 2014. Automatic question generation for educational applications – the state of art. In *Advanced Computational Methods for Knowledge Engineering*, pages 325–338.
- Seungyeon Lee and Minho Lee. 2022. Type-dependent prompt CycleQAG : Cycle consistency for multi-hop question generation. In *Proc. International Conference on Computational Linguistics*, pages 6301–6314.
- 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 *Proc. Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.
- F.M. Lord. 1980. *Applications of item response theory to practical testing problems*. Routledge, Evanston, IL, USA.
- 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. 2022. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proc. Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.
- Manav Rathod, Tony Tu, and Katherine Stasaski. 2022. Educational multi-question generation for reading comprehension. In *Proc. Workshop on Innovative Use of NLP for Building Educational Applications*, pages 216–223.
- Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proc. Conference on Empirical Methods in Natural Language Processing*, pages 193–203.
- Anirudh Srikanth, Ashwin Shankar Umasankar, Saravanan Thanu, and S. Jaya Nirmala. 2020. Extractive text summarization using dynamic clustering and coreference on BERT. In *Proc. International Conference on Computing, Communication and Security*, pages 1–5.
- Megha Srivastava and Noah Goodman. 2021. Question generation for adaptive education. In *Proc. Annual Meeting of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing*, pages 692–701.
- Maomi Ueno and Yoshimitsu Miyazawa. 2018. IRT-based adaptive hints to scaffold learning in programming. *IEEE Transactions on Learning Technologies*, 11(4):415–428.



- Asahi Ushio, Fernando Alva-Manchego, and Jose Camacho-Collados. 2022. Generative language models for paragraph-level question generation. In *Proc. Conference on Empirical Methods in Natural Language Processing*.
- Masaki Uto, Itsuki Aomi, Emiko Tsutsumi, and Maomi Ueno. 2023. Integration of prediction scores from various automated essay scoring models using item response theory. *IEEE Transactions on Learning Technologies*, pages 1–18.
- Masaki Uto and Maomi Ueno. 2020. A generalized many-facet Rasch model and its Bayesian estimation using Hamiltonian Monte Carlo. *Behaviormetrika*, 47(2):469–496.
- Wim J. van der Linden and Cees A.W. Glas. 2010. *Elements of Adaptive Testing*. Springer New York.
- Jianxing Yu, Qinliang Su, Xiaojun Quan, and Jian Yin. 2023. Multi-hop reasoning question generation and its application. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):725–740.
- 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*, 40(1):1–43.

# Evaluating Classroom Integration for Card-it: Digital Flashcards for Learning Italian Morphology

**Mariana Shimabukuro**

Ontario Tech University, Canada

mariana.shimabukuro@ontariotechu.ca

**Jessica Zipf**

University of Konstanz, Germany

jessica.zipf@uni-konstanz.de

**Shawn Yama**

Ontario Tech University, Canada

shawn.yama@ontariotechu.net

**Christopher Collins**

Ontario Tech University, Canada

christopher.collins@ontariotechu.ca

## Abstract

This paper presents Card-it, a web-based application for learning Italian verb conjugation. Card-it integrates a large-scale finite-state morphological (FSM) analyzer and a flashcard application as a user-friendly way for learners to utilize the analyzer. While Card-it can be used by individual learners, to support classroom adoption, we implemented simple classroom management functionalities such as sharing flashcards to a class and tracking students' progression. We evaluated Card-it with teachers of Italian. Card-it was reported as engaging and supportive, especially by featuring two different quiz types combined with a verb form look-up feature. Teachers were optimistic about the potential of Card-it as a classroom supplementary tool for learners of Italian as L2. Future work includes sample sentences and a complete learners evaluation.

## 1 Introduction

Learning verb morphology plays a crucial role in the acquisition of morphologically rich languages (Slabakova, 2009), such as Italian and French. Thus, learners of Italian deal with the acquisition of a rich system of verbal inflections (e.g., Pizzuto and Caselli, 1994). Explicit morphological instructions and training have been shown to help students on acquiring new words as well as to improve their syntactic knowledge (Chen and Schwartz, 2018; Mobaraki and Jahromi, 2019). Similarly, raising meta-linguistic awareness improves the learners' production and competence in second language (L2) acquisition (Heift, 2004; Kieseier et al., 2022). To support learners of Italian as L2, we designed, implemented, and evaluated Card-it with the help of experts: teachers of Italian as a foreign language. Card-it fosters meta-linguistic knowledge when presenting linguistic information on the analysis of verb forms (i.e., for the verb *mangiare* (to eat) “Prima Persona Singulare Presente Indicativo” → (*io mangio*) along with additional

explanations of linguistic categories related to verb morphology that are displayed on demand. In addition, meta-linguistic information is also used to present corrective feedback (see Sec. 4.2).

Card-it is an online application for teachers and learners of Italian to create collections of digital flashcards – based on a semi-automatic approach – with which they can study and test themselves on verb morphology explicitly. Our choice for using a digital flashcard design reflects a traditional way of learning vocabulary explicitly, which has been shown to be a successful learning method that is perceived well by students (Yüksel et al., 2022). While some flashcard systems may support verb morphology with pre-defined cards and modules, they do not allow for the customization of cards or decks (e.g., Memrise<sup>1</sup>). Other systems support custom card collections, but they require manual input of the card information (e.g., Anki<sup>2</sup>). Yet, these systems do not enable teachers to track and analyze their students' progress over time. In addition, Card-it's learner-centred design embeds corrective feedback, meta-linguistic information, and different study modes.

This paper introduces the system's architecture, the FSM implementation, and Card-it's iterative design and features. Lastly, we report the results of a brief evaluation with Italian teachers which indicates Card-it's potential for their classroom and outlines our future steps towards a learners evaluation.

## 2 Related Work

Traditionally, Natural Language Processing (NLP) tools like an FSM are a component of larger pipelines, for example, as a tokenizer (e.g., Jurafsky and Martin, 2009). As a result, using these tools is often not intuitive or easy for users unfamiliar with NLP. However, since these tools can

<sup>1</sup><https://www.memrise.com/>. Accessed 05-2023.

<sup>2</sup><https://www.ankiapp.com/>. Accessed 05-2023.

work with text, NLP has become an integral part of the field of Computer-Assisted Language Learning (CALL), with several systems using NLP tools in a language-learning context. Examples include E-Tutor (Heift, 2010), an intelligent tutoring system for learners of German that is fully incorporated into the German curriculum at Simon Fraser University; TAGARELA (Amaral and Meurers, 2011), a system for Portuguese that includes exercises on vocabulary; and FeedBook (Meurers et al., 2019), an intelligent tutoring system for English that can be fully integrated into regular classes.

Similarly, Google-Assisted Language Learning (GALL), corpus-based or data-driven learning (DDL) are increasing in popularity as language learning tools (Conroy, 2010; Pérez-Paredes, 2022). While GALL refers specifically to learners using tools provided by Google, both GALL and DDL happen when learners take advantage of online access and text processing power to use corpus tools, such as dictionaries and linguistic corpora.

Furthermore, Yoon (2016) verified that DDL was an effective cognitive tool for helping people with their lexical and grammatical problems while dealing with concordance tasks; for example, learning frequent word pairs such as *to take* instead of *to eat* a [medicine] *pill*. However, he suggests that some of the available resources are not user-friendly and difficult to use, such as functions for linguistic resources applied for stemming. That said, Card-it’s design uses a learner-centred approach with teacher support features; it provides a user-friendly interface to leverage an FSM to power a semi-automatic generation of flashcards that can be used to study and self-assess Italian verb conjugations. Related to using FSM in Card-it, Kaya and Eryiğit (2015) used a Finite-State Transducer to power a Turkish word synthesis system and a word-level translation system between Turkish and English. Another example is the ICALL system for two Saami languages that is based on Finite-State Transducers (Antonsen et al., 2013).

### 3 Card-it: System Architecture

Card-it is a web-based application consisting of two components: back-end and front-end.

**Back-end: The FSM Analyzer.** The main component of the back-end is our FSM, containing over 5000 verb lemmata and their conjugations Beesley and Karttunen (2003). It was created by extracting verb roots from free resources, the *Morph-it!* lexi-

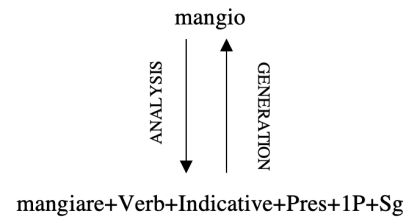


Figure 1: Example of FSM Analysis and Generation for the verb *mangio* “I eat”.

con by Zanchetta and Baroni (2005) and the online dictionary provided by one of Italy’s leading news magazines, *Corriere della Sera*<sup>3</sup>. FSMs are usually part of a text processing pipeline within NLP tools. Here, we leveraged our FSM as a dynamic form generator and analyzer in a language-learning context. The FSM ties a verb form to its linguistic analysis: it may analyze a verb form and return its linguistic tags (analysis) or generate a verb form given its linguistic tags (generation) – see Fig. 1.

In our case, the FSM consists of a lexicon that contains verb stems, their inflectional paradigms and the appropriate morphological analysis. The lexicon of the FSM creates all verb forms following the regular pattern of concatenating stems with their respective inflectional endings. With the use of regular expressions the FSM is able to manipulate those regular forms of the lexicon on the basis of phonological rules. For example, some forms require the insertion of an *-h* to retain certain pronunciation patterns. Consider the verb *mancare* (“to miss”): the regular inflection paradigm in the lexicon creates the incorrect form *manci* (“you miss”), for the second person singular present indicative. However, to retain the correct pronunciation, the correct form is *manchi*. Whenever the FSM is run, it first creates all forms in the lexicon and then applies regular expressions to manipulate these forms based on phonological rules of the language. This architecture allows us to build a powerful and large morphological resource since it automatically creates verb forms on the basis of their stems. If we were to add new verbs to our tool, it simply requires to manually add verb stems into the FSM lexicon.

Verbs generated by the FSM, user accounts, flashcards and classroom organization are stored in a MySQL database. A Flask middleware is responsible for querying changes users request from the front-end. These changes are related to flash-

<sup>3</sup>[https://dizionari.corriere.it/dizionario\\_italiano/](https://dizionari.corriere.it/dizionario_italiano/). Accessed 05-2023

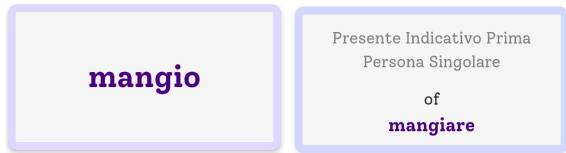


Figure 2: Both sides of the flashcard corresponding to the verb form *mangio* “I eat”. Side 1: Verb form (left); and Side 2: morphological information (right).

card, classroom, and account organization. The main advantage of this back-end architecture is to scale the system for multiple users simultaneously; this integration approach has been taken by others (de Bernardinis et al., 2015). A set of Python scripts are responsible for parsing and updating the database with any changes to the FSM; currently, these updates are triggered manually whenever the list of verbs or morphology is altered.

**Front-end: User Interface.** The user interface front-end of Card-it is developed with React.js. The main function of the front-end is the flashcard design for users to study and be assessed from. Sec. 4 explains Card-it’s digital flashcards design and interaction.

## 4 Card-it Design and Features

Card-it can be used for autonomous learners who may interact with the app to study Italian conjugations on their own. In addition, Card-it can also be integrated by teachers in the classroom. In either case, learners interact with verbs and conjugations via digital flashcards.

### 4.1 Grouping and Organizing Flashcards

The flashcards reflect a traditional way of language learning. Particularly, the flashcard design reflects both directions of the FSM: one side of the card contains a verb form, the other its linguistic attributes (compare Figs. 1 and 2); learners may choose which side they want to use for studying.

Flashcards can be organized in decks; decks can be organized in collections. Both learners and teachers can organize flashcards according to their learning or teaching needs. For example, a teacher may create collections for different language classes: in a collection “Italian for Beginners”, the teacher may add a deck for present tense only, another for past tense(s), and so on.

Users can create decks of cards by searching the database for specific verbs and filtering values for the categories tense, mood, number, and person.

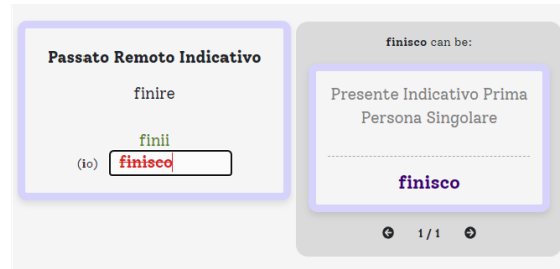


Figure 3: On the left side’s card, see an example of the “Conjugation” task for studying or quizzing. Corrective Feedback is displayed within the card for both studying or testing with this task. On the right side, see a panel with feedback about the incorrect verb form input *finisco*, helping learners to recall the possible conjugations for *finisco* – available for the quiz version only.

Alternatively, if no value is chosen, Card-it returns all forms for that category. E.g., one may search the verb *amare* “to love”, selecting the values *present tense* and *indicative mood*, but selecting none for person and number. Card-it returns 1<sup>st</sup>, 2<sup>nd</sup> and 3<sup>rd</sup> person singular and plural forms of *amare*, where each form is a flashcard. Users can select any flashcards they want to add to a specific deck.

The knowledge of the underlying linguistic concepts benefits the acquisition of a new language Heift (2004). Therefore, we made the decision to include the morphological attributes in the application to raise meta-linguistic awareness. Card-it also offers a page with definitions and explanations of all the terms used (i.e., “What is tense?”).

### 4.2 Studying and Self-Assessing

Card-it offers different study modes and ways to interact with its flashcards.

**Studying with Card-it.** One way is to use the **flip card** functionality, where Card-it presents the user with one side, and the learner can think about the content on the corresponding side. When hovering the mouse over the card, the flashcard flips to its other side, and learners can check their answer. Another mode is **conjugation**. Here, the flashcard presents the user with the infinitive form of a verb, a tense/mood combination, and personal pronouns for number/person configurations and prompts the user to type in the corresponding verb form. If wrong, the system returns the corrected answer as seen on the left side of Fig. 3, showing the “conjugation” study mode, with corrective feedback.

**Self-assessment and corrective feedback.** For testing, Card-it has two different types of quizzes, called **Identify Tense**, **Conjugate**, and a third

**Mixture**, a random mix of tasks from the other two types. While “Conjugate” corresponds to the above-described study mode prompting the user to type in the corresponding conjugated form, in quiz mode, it additionally contains a “Hint” button that displays multiple choice options when used (Fig. 4), otherwise hidden by default.

Studies have shown the importance of informative feedback for a positive learning trajectory as it helps learners to understand the nature of their mistakes and to improve in the future (e.g., Heift, 2004). Card-it returns informative feedback to the learner by checking whether their incorrect answer corresponds to another morphological analysis and returning that information to them, see Fig. 3. The second quiz type, “Identify Tense”, presents learners with a specific verb form, asking them to select its respective tense (Fig. 4). All quiz types may be used for self-assessment or as classroom activities.

### 4.3 Classroom Management and Analytics

To enable classroom and teacher support, we focused on 3 main tasks. The tasks supported in this category are (1) creating classrooms and generating a unique code that is shared with students allowing them to join it, (2) sharing specific collections to one or multiple classrooms, and (3) tracking the progress of students enrolled in the classroom.

After students join the classroom using the code, they can explore all collections and decks their teacher shares. Similarly, students have access to both studying and quiz modes for all decks in the classroom. Teachers can access statistical information on the students’ progress with the classroom decks. Teachers can analyze individual attempts for each student with a breakdown of correct and incorrect answers. Alternatively, teachers can see average scores per attempt for the entire group; and analyze the class’ progress over time. Lastly, Card-it shows the number of correct attempts for each card in a deck. Thus, the teacher can pinpoint the specific cards students had the most trouble with.

## 5 Evaluation

We took an iterative design approach for implementing Card-it, where we performed a preliminary expert evaluation ( $N = 2$ ) with teachers of Italian at the Institute of Speech and Language at our university with an earlier version of the application. Based on this preliminary evaluation, we determined the fitness of the flashcards and the

quiz formats and iterated over the application. The teachers responded positively to Card-it as a digital version of their current classroom practices, such as verb conjugation worksheets. We also learned that Card-it could be adopted as a supplemental tool to the classroom, which led us to implement the classroom features. The following section describes our second expert evaluation.

### 5.1 Card-it Expert Evaluation

After implementing changes to reflect the feedback from the early preliminary evaluation; we reached out to Italian teachers via our professional networks. In total, 9 teachers from 2 institutions in Germany were invited to participate. Of those, 5 volunteered, but only 3 completed the study. Participants were teachers of Italian language courses; after receiving the study instructions, they had 14 days to follow to complete all steps remotely, then compensated with a \$20 Amazon gift card.

#### 5.1.1 Methodology

We ran our expert evaluation remotely, which allowed us to provide flexibility to participants to complete the study. Participants were asked to follow three steps to complete the study: (1) Watch a recorded video demo of Card-it’s main features; (2) Explore Card-it on their own using both teacher and student account types; (3) Answer a survey questionnaire about their experience using Card-it. In the survey, we asked 5-Point Likert Scale questions on general usability, the potential for classroom adoption, and specific questions on different features such as studying and testing modes. We also asked experts to answer a section where they give their opinions from a student perspective.

#### 5.1.2 Results and Discussion

The system’s usability was rated positively, with two experts selecting *easy* and one expert *very easy*. All experts rated both quiz types, “Conjugate” and “Identify Tense”, as either *appropriate* or *very appropriate*. One expert mentioned the quizzes were their favourite features. When asked to rate the classroom management usability, two chose *good* and one *very good*. As a follow-up, we asked them about the steps to create a classroom: one expert found it *difficult*, and the others *easy*. They all mentioned that they could foresee themselves using Card-it for homework in their classes or as a tool for students to self-study at home. When asked to take on a student’s perspective, they all rated the



Figure 4: On the left, an example of the “Hint” button used during the “Conjugate” quiz gives the learner the option to select the correct verb form from one of the given choices to the right instead of typing it in. On the right is an example of the quiz type “Identify Tense”; the learner selects the correct tense from the choices given.

quiz and verb look-up features of Card-it *most useful*. Yet, they suggested including translations and example sentences containing the individual verbs as it would be useful for students and teachers’ perspectives.

## 6 Future Work and Conclusion

This paper discussed the power of the adequate use of NLP tools in language learning, including designing appropriate interfaces. We presented Card-it as a user-friendly app for learning Italian verb conjugation using digital flashcards; we also described Card-it’s classroom management and analytics features (more details in App. A). Lastly, we discussed our iterative approach to design, which combined expert evaluations between iterations of Card-it. The results of the expert evaluation show that according to their expertise, Card-it is an appropriate conjugation tool for autonomous learning and for classroom integration as a supplementary resource. Card-it’s usability and different quiz functions were positively evaluated. Nonetheless, we also learned that Card-it might be further improved by adding example sentences. The most promising result from the evaluation is the experts’ expression of interest in using Card-it in their classrooms.

Despite asking for the experts’ perspectives as students, it would be more reliable to run a user study with learners of Italian as a second language. We are designing a remote longitudinal study with 3 weekly sessions. At the end of each session, participants are invited to submit a Card-it quiz and a short usability survey. We also plan on testing their knowledge of a set of verb conjugations before and after their study period of 3 weeks. Other future directions may include gamification of Card-it’s quizzes and quiz modes that can support live classroom exercises such as Kahoot (Dellos, 2015).

## Acknowledgements

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

## References

- Luiz A. Amaral and Detmar Meurers. 2011. [On using intelligent computer-assisted language learning in real-life foreign language teaching and learning](#). *ReCALL*, 23(1):4–24.
- Lene Antonsen, Ryan Johnson, Trond Trosterud, and Heli Uiho. 2013. Generating modular grammar exercises with finite-state transducers. In *Proceedings of the second workshop on NLP for computer-assisted language learning at NODALIDA 2013.*, pages 27–38.
- Kenneth R. Beesley and Lauri Karttunen. 2003. *Finite State Morphology*. CSLI Publications, Stanford.
- Xi Chen and Mila Schwartz. 2018. [Morphological awareness and literacy in second language learners: a cross-language perspective](#). *Reading and Writing*, 31(8):1685–1694.
- Mark A Conroy. 2010. Internet tools for language learning: University students taking control of their writing. *Australasian Journal of educational technology*, 26(6).
- Jacopo de Bernardinis, Carlo Castagnari, and Giorgio Forcina. 2015. [ICALL-IT : an Intelligent Computer-Assisted Language Learning Platform for the Italian language ICALL-IT : an Intelligent Computer-Assisted Language Learning](#). (March 2016).
- Ryan Dellos. 2015. Kahoot! a digital game resource for learning. *International Journal of Instructional technology and distance learning*, 12(4):49–52.
- Trude Heift. 2004. [Corrective feedback and learner uptake in CALL](#). *ReCALL*, 16(2):416–431.
- Trude Heift. 2010. [Developing an intelligent language tutor](#). *CALICO Journal*, 27(3):443–459.

- Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.*, 2 edition. Prentice Hall, Upper Saddle River.
- Hasan Kaya and Gülşen Eryiğit. 2015. Using Finite State Transducers for Helping Foreign Language Learning. In *Proceedings of the 2nd Workshop on Natural Language Processing Techniques for Educational Applications*, pages 94–98, Beijing.
- Teresa Kieseier, Dieter Thoma, Markus Vogelbacher, and Hopp Holger. 2022. Differential effects of metalinguistic awareness components in early foreign language acquisition of english vocabulary and grammar. *Language awareness*, 31(4):495–514.
- Detmar Meurers, Kordula De Kuthy, Florian Nuxoll, Björn Rudzewitz, and Ramon Ziai. 2019. [Scaling Up Intervention Studies to Investigate Real-Life Foreign Language Learning in School](#). *Annual Review of Applied Linguistics*, 39:161–188.
- Mahmoud Mobaraki and Abolfazl Mosaffa Jahromi. 2019. [Morphological knowledge and learning a foreign language: A case study](#). *Journal of Linguistics and Literature*, 3(1):1–4.
- Pascual Pérez-Paredes. 2022. A systematic review of the uses and spread of corpora and data-driven learning in call research during 2011–2015. *Computer Assisted Language Learning*, 35(1-2):36–61.
- Elena Pizzuto and Maria Cristina Caselli. 1994. [The Acquisition of Italian Verb Morphology in a Cross-Linguistic Perspective](#). In Yonata Levy, editor, *Other Children, Other Languages. Issues in the theory of Language Acquisition*, pages 137–188. Psychology Press, New York.
- Roumyana Slabakova. 2009. [Features or parameters: which one makes second language acquisition easier, and more interesting to study?](#) *Second Language Research*, 25:313–324.
- Choongil Yoon. 2016. Concordancers and dictionaries as problem-solving tools for esl academic writing. *Language Learning & Technology*, 20(1):209–229.
- Hatice Gülru Yüksel, H. Güldem Mercanoğlu, and Mihriban Betül Yılmaz. 2022. [Digital flashcards vs. wordlists for learning technical vocabulary](#). *Computer Assisted Language Learning*, 35(8):2001–2017.
- Eros Zanchetta and Marco Baroni. 2005. Morph-it! a free corpus-based morphological resource for the italian language. *Corpus Linguistics 2005*, 1(1).

## A Classroom Management

Fig. 5 shows an example classroom with two collections and the entry code for students to join the classroom:



Figure 5: Example of a classroom with two collections and its entry code.

Fig. 6 shows the statistical overview of students' performance in a quiz. Teachers may filter for a specific collection (here: Presente Indicativo), deck (here: Regular Verbs) and quiz type (here: Conjugate). Additionally, teachers see the score for each student:

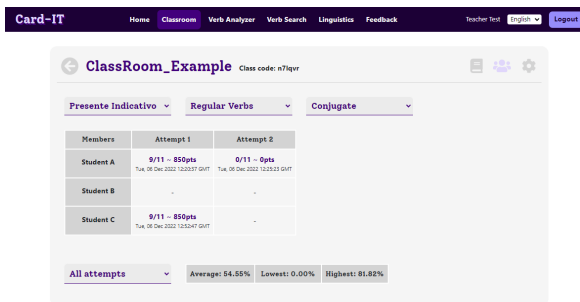


Figure 6: Example of a statistical performance overview.

Fig. 7 illustrates how teachers can check on the groups' performance on every single card, sorted from the least correct to the most correct:

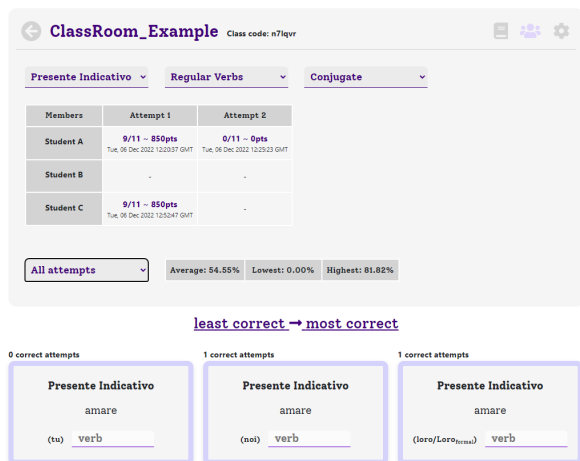


Figure 7: Example of a statistical performance overview.

Teachers may select one particular student to get detailed information on their performance, as in Fig. 8:

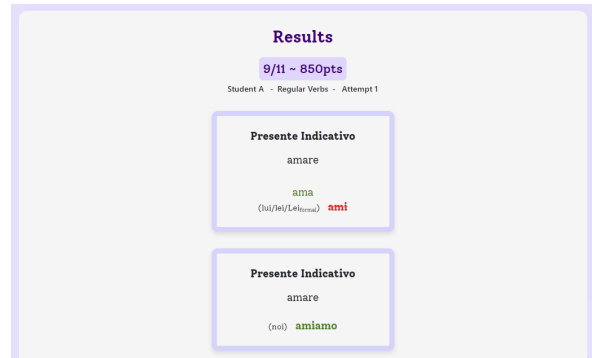


Figure 8: Example of a detailed performance overview of a particular learner.

Fig. 9 shows the same example classroom as in Fig. 5 but from the students' perspective. Here, students can select one of the three quiz types or scroll down for study mode:

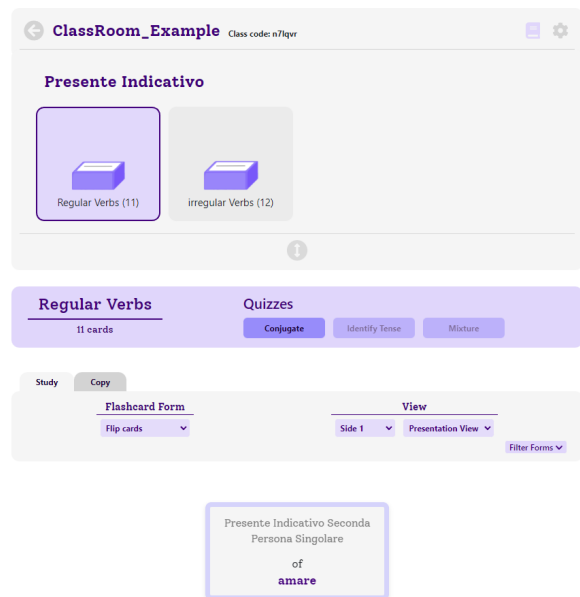


Figure 9: Example of a classroom as seen in a learner's account.



# Scalable and Explainable Automated Scoring for Open-Ended Constructed Response Math Word Problems

Scott Hellman and Alejandro Andrade and Kyle Habermehl

Pearson  
Boulder, CO

{scott.hellman, alejandro.andradelotero,  
kyle.habermehl}@pearson.com

## Abstract

Open-ended constructed response math word problems ("math plus text", or MPT) are a powerful tool in the assessment of students' abilities to engage in mathematical reasoning and creative thinking. Such problems ask the student to compute a value or construct an expression and then explain, potentially in prose, what steps they took and why they took them. MPT items can be scored against highly structured rubrics, and we develop a novel technique for the automated scoring of MPT items that leverages these rubrics to provide explainable scoring. We show that our approach can be trained automatically and performs well on a large dataset of 34,417 responses across 14 MPT items.

## 1 Introduction

Math word problems are a common question type in both formative and summative mathematics assessment. In a math word problem, the prompt describes a scenario and asks the student to calculate some value or construct some mathematical expression pertaining to that scenario. Such problems assess both the student's ability to carry out mathematical computation and reasoning as well as their ability to apply their knowledge in determining how to solve a mathematical problem.

Automated assessment of closed constructed-response (CR) math problems is straightforward, although complexities arise due to the variety of possible representations for a given mathematical expression. Examples of automated assessment systems for closed CR items include m-rater (Fife, 2017) and MathQuery (Streeter et al., 2011). In contrast, open-ended CR math problems are difficult to automatically score, since responses to open CR items combine mathematical expressions with prose explanations. And if a problem asks students to both compute a value and explain their computation, that introduces the complexity of partial

credit; in the dataset we consider in this work, score ranges for items vary between 0–2 and 0–4. Even for humans, these sorts of items, with partial credit and open-ended responses, are time-consuming to score (Stankous, 2016).

Automated assessment of CR items outside of mathematics is now common, thanks to the achievements of researchers in the areas of Automated Essay Scoring (AES) and Automated Short Answer Scoring (SAS). The reliability of AES systems is often comparable to that of humans (Shermis and Burstein, 2003, 2013), and the same is true for SAS systems (Butcher and Jordan, 2010). Given that MPT items are themselves CR items, this suggests that such approaches could also be used for MPT; research in this area is promising, but sparse (Erickson et al., 2020; Cahill et al., 2020).

How mathematical expressions are encoded in response text is a key attribute of a given MPT dataset. In this work, we use data generated by a writing environment that allows students to enter mathematics using a math editor tool. Any math written in this tool is represented in the final response text as Content MathML (an XML-based specification for the representation of mathematics). As students can also write math outside of the math editor, the dataset that we consider in this work contains math represented both in MathML and in plain text, often within the same response.

Given this set of challenges, our interest is in creating an *explainable* predictive model for MPT. Such a model would be able to differentiate, for example, between a response that received a 1 out of 3 because it contained the correct final answer without showing work, and a response that received a 1 out of 3 because it contained correct reasoning but incorrect computations. A model that successfully achieved this would be useful both for students, as they would better understand why their responses received their assigned scores, as well as for test administrators, as the explanations would build trust

in the validity of the model’s scoring.

This paper is structured as follows. We begin with a discussion of related work and a detailed description of our task. We introduce a novel scoring model that uses the rubric’s structure to provide explainable scoring for MPT, and show how our model can be automatically trained. We then present experimental results that show the effectiveness of our approach, and conclude with a discussion of the present and future work.

## 2 Related Work

There is a substantial literature around the automated scoring of non-mathematical CR items. Work on AES dates back to the 1960s (Page, 1966), and modern-day AES systems involve a wide variety of approaches, including linear regression (Larkey, 1998), random forests (Hellman et al., 2019), and neural networks (Taghipour and Ng, 2016; Dong et al., 2017; Riordan et al., 2017). Short answer scoring is also relevant, as our MPT responses tend to be only a few hundred characters long. For SAS, many systems involve paraphrase detection, or some similar notion of semantic similarity to reference answers (Leacock and Chodorow, 2003; Tandalla, 2012; Ramachandran et al., 2015; Kumar et al., 2017).

While much work has been done on AES and SAS, as well as around the automated *solving* of math word problems (e.g. Kushman et al. 2014; Huang et al. 2016; Wang et al. 2017; Xie and Sun 2019), work around the automated *scoring* of math word problems is more limited. Livne et al. demonstrate a system that successfully uses instructor-provided reference answers to automatically score responses to closed CR math word problems (Livne et al., 2007). Lan et al. present a system that predicts scores by embedding multi-step math responses using a bag-of-expressions model, a bag-of-words approach designed to capture mathematical features (Lan et al., 2015). Once embedded, they use a combination of clustering and limited human scoring to score all responses. However, while their items were open CR math word problems, any prose in student responses was ignored by the scoring system.

Some systems do attempt to grapple with the full complexity of open CR math word problems. Kadupitiya et al. present a system that can score CR math word problems for summative assessments whose responses contain both prose and

math (Kadupitiya et al., 2017). Their system assumes that all math is encoded as MathML, and prose is handled by estimating the semantic similarity of response phrases to known reference phrases. Erickson et al. (Erickson et al., 2020) investigated the effectiveness of random forests, XGboost, and LSTMs for scoring formative open CR math problems with only plain text responses, and follow-up work has shown that transformer-based approaches can also perform well on this task (Baral et al., 2021; Shen et al., 2021).

As mentioned above, we expect that many real-world MPT datasets will include responses that contain math represented both as plain text and as MathML. To the best of our knowledge, Cahill et al. is the only published work that attempts to score these sorts of responses (Cahill et al., 2020). In their work, they extract plain text math from student responses using regular expressions, and then use the m-rater (Fife, 2017) math scoring system to evaluate the correctness of this extracted math. They then build a feature space that includes binary features indicating whether certain rubric elements were covered by the student response. By training machine learning models on this feature space, they create models with interpretable features. This process requires knowledge of the rubric during training. Our work differs from Cahill et al. in that the model that we introduce relies *only* on features that are aligned with the rubric, and produces scores that are inherently explainable. Furthermore, it requires no knowledge of the rubric during training. We also evaluate our approach across a wider variety of items with more responses per item.

## 3 Open Constructed Response Math Word Problems

The dataset we use in this work is proprietary, so we have adapted an item from the GSM8K dataset<sup>1</sup> (Cobbe et al., 2021) as an illustrative example, shown in Table 1. In this example, the prompt establishes a scenario and asks the student to compute a value related to that scenario. The rubric defines three binary components that a response can achieve, which defines the score range for this item to be from 0 to 3. Finally, the example response shows a typical mixing of MathML and prose.

<sup>1</sup>Dataset located at [https://github.com/openai/grade-school-math/tree/master/grade\\_school\\_math/data](https://github.com/openai/grade-school-math/tree/master/grade_school_math/data).

Example Prompt	Albert is wondering how much pizza he can eat in one day. He buys 2 large pizzas and 2 small pizzas. A large pizza has 16 slices and a small pizza has 8 slices. If he eats it all, how many pieces does he eat that day?
Example Rubric	1 point for correct computation (48 pieces). 1 point for correct modeling of the number of slices for the large pizza ( $2 * 16$ ) and the small pizza ( $2 * 8$ ). 1 point for correct modeling of the total number of slices ( $32 + 16$ ).
Example Response	He eats 32 from the largest pizzas because $\langle \text{math} \rangle \langle \text{apply} \rangle \langle \text{eq} \rangle \langle \text{apply} \rangle \langle \text{times} \rangle \langle \text{cn} \rangle 2 \langle \text{cn} \rangle \langle \text{cn} \rangle 15 \langle \text{cn} \rangle \langle \text{apply} \rangle 32 \langle \text{apply} \rangle \langle \text{math} \rangle$ . He eats 16 from the small pizza because $\langle \text{math} \rangle \langle \text{apply} \rangle \langle \text{eq} \rangle \langle \text{apply} \rangle \langle \text{times} \rangle \langle \text{cn} \rangle 2 \langle \text{cn} \rangle \langle \text{cn} \rangle 8 \langle \text{cn} \rangle \langle \text{apply} \rangle 16 \langle \text{apply} \rangle \langle \text{math} \rangle$ . He eats 48 pieces because $\langle \text{math} \rangle \langle \text{apply} \rangle \langle \text{eq} \rangle \langle \text{apply} \rangle \langle \text{plus} \rangle \langle \text{cn} \rangle 32 \langle \text{cn} \rangle \langle \text{cn} \rangle 16 \langle \text{cn} \rangle \langle \text{apply} \rangle 48 \langle \text{apply} \rangle \langle \text{math} \rangle$ .
Normalized Response	He eats 32 from the largest pizzas because $2 * 15 = 32$ . He eats 16 from the small pizza because $2 * 8 = 16$ . He eats 48 pieces because $32 + 16 = 48$ .

Table 1: Example item and response. Adapted from the GSM8K dataset (Cobbe et al., 2021)

We are focused on word problems that ask the student to construct some mathematical equation and/or compute some number, as well as to provide the work and reasoning that they used in coming to their answer. For some items, this explanation is required to be prose, while for others the chain of mathematical expressions that led to the answer can suffice.

Each item has a rubric composed of some number of *computation*, *modeling*, and *reasoning* components, each of which is worth one point. Computation components generally refer to the presence of a correct final answer, modeling components to showing the correct mathematical derivation of the final result, and reasoning components to an explanation of why those steps were taken. A given rubric may not include all three of these components, and may also define multiple components of a given kind. The final score of a response is the sum of these binary component scores. Note that even if a rubric does not require a prose explanation, the student may still include prose in their final response.

The characteristics of the dataset used in this work are shown in Table 2. A critical aspect of our dataset is that MPT problems are, in general, quite difficult for students to answer correctly. For some items, more than 70% of student responses received a score of 0. This is an expected feature of our dataset, as math word problems are known to be substantially harder for students to solve than conventional math problems (Cummins et al., 1988).

Student responses are written in an environment

that supports the entry of both plain text in a conventional text field and of math via a math editor. Critically, arbitrary text input is allowed in the math editor, to support the presence of variables in the student answer. While the expectation is that students will use this math editor to write the relevant mathematical expressions, and write the rest of the response outside of the math editor, in practice students often write prose inside of the math editor and math expressions outside of the math editor. Thus, we cannot look only at the MathML in a response to identify the mathematical statements produced by the student, and we cannot look only at the plain text to identify their explanations and supporting arguments. Because of this, we believe the best way to score MPT responses is by converting them to a normalized form.

This normalization process consists of three steps: first, we convert mathematical terms in the response into their symbolic equivalents, e.g. "eight" to "8", or "plus" to "+". Next, we need to account for prose written in the math editor. We identify MathML containing chains of variables being multiplied together that appear to spell out English words. When such a chain is found, it is removed from the MathML and converted to plain text by preserving the order of the variables and removing the multiplication operators. This replaces the variables in the MathML by their corresponding plain text word. Finally, we transform all remaining MathML into plain text by taking the in-order traversal of the expression tree defined by the MathML.

item	grade	domain	response count	mathml %	char count	score range	sp 0 %
1	7	algebra	4095	18.1	167	0-2	37.2
2	high school	algebra	2634	26.4	166	0-2	77.4
3	high school	algebra	2472	47.3	109	0-2	92.6
4	high school	algebra	2362	34.4	97	0-2	81.8
5	high school	algebra	5701	28.3	202	0-4	70.0
6	4	arithmetic	1266	72.5	87	0-2	50.0
7	5	geometry	1596	62.7	125	0-3	32.7
8	7	algebra	1665	29.6	198	0-3	41.4
9	7	algebra	1085	32.7	95	0-2	39.9
10	7	algebra	838	30.8	91	0-3	70.5
11	high school	algebra	1581	19.7	294	0-2	63.5
12	high school	algebra	1495	22.2	294	0-4	72.5
13	6	algebra	6018	36.0	212	0-3	54.6
14	6	arithmetic	1609	29.5	259	0-4	40.1

Table 2: Dataset summary. Mathml % is the mean percentage of characters in a response occurring inside of MathML spans. Char count is the mean number of characters. Sp 0 % is the percentage of responses at scorepoint 0.

## 4 Explainable Scoring

As outlined in Section 3, the rubrics for our MPT items are highly structured. We leverage this structure to create a new approach to the automated scoring of MPT items by essentially codifying the rubric in a machine-understandable way. The close alignment of our model with the rubric produces predictions that are inherently explainable.

*Rules* form the core building block of our approach. Rules encode short mathematical expressions and the transformations required to convert them into other lexically distinct but semantically identical forms. For example, a rule encoding " $2 + 3$ " could generate " $3.0 + 2$ " as an alternative form. These alternate forms account for different mathematical properties, principally commutativity and conversion between floats and integers (for whole numbers). To account for variables, we also allow single letters to serve as operands in our expressions.

To determine if a rule is present in a student response, we first extract all mathematical text from the normalized text of the response. This is to prevent superfluous words from obscuring the underlying mathematics. See Figure 1b for an example. Then, if any of the forms of a rule are present as a substring of the extracted math, that rule is considered to be present in the response.

The amount of prose in a response is highly item-dependent. To account for items where prose is important, we also include the ability to write regu-

lar expressions as rules. Such a rule is found in a response if its constituent regular expression has at least one match in the response.

Assembling these rules into a form that can automatically score responses is done as follows. We define a *group* to be a list of rules, and we consider a group to be present in a response if any of its constituent rules are present. This allows us to capture mathematics that are equivalent under the rubric but not captured by the lexical transformations of our rules, for instance, " $2 * 16$ " and " $16 + 16$ " could be two valid ways of writing an expected expression.

We then create *evidence* out of these groups. Evidence is a list of groups, and we consider evidence to be present in a response if *all* of its constituent groups are in the response. This allows us to capture rubric elements that require the student to cover multiple areas. For example, if a student needs to show two distinct values to achieve a Computation component, we can capture this notion by constructing evidence with two groups, one for each of those two distinct values.

Finally, to mirror the structure of the rubric components, we collect evidence into *scorable traits*. A scorable trait contains lists of positive and negative evidence. If any positive evidence and no negative evidence is present in a response, then the scorable trait scores a 1. Otherwise, it scores a 0. We include this concept of negative evidence to account for misconceptions and other incorrect mathemat-

```

{
  "name": "Modeling",
  "positive_evidence": [
    {
      "name": "Correct equations",
      "equations": [
        ["2*16", "16+16"],
        ["2*8", "8+8"]
      ]
    }
  ]
}

```

(a) Example Scorable Trait

```

Normalized Response:
He eats 32 from the largest pizzas because
2*15=32. He eats 16 from the small pizza
because 2*8=16. He eats 48 pieces because
32+16=48.

Extracted Math:
32 2 * 15 = 32 16 2 * 8 = 16 46 32 + 16 = 48

Explanation:
• Scorable Trait "Modeling" scored 0
  • Evidence "Correct equations" not found
    • Group ["2*16", "16+16"] not found

```

(b) Example Response

Figure 1: An example scorable trait is shown in Figure 1a. This scorable trait captures a modeling component from the example shown in Table 1. This scorable trait is composed of one piece of positive evidence, which in turn consists of two groups. The first detects if the student found a correct equation for the number of slices for the large pizza. The second detects if the student found a correct equation for the number of slices for the small pizza. Figure 1b shows the normalized response from Table 1, alongside the math extracted from the response. The highlighted characters indicate where in the response the rules from the Scorable Trait were found. The automatically generated explanation of the score is also shown.

ics that can prevent a student from receiving full credit on a rubric component. For example, if an item asked the student to compute 4 divided by 2, the student could incidentally compute the correct value by subtracting 2 from 4.

We construct a number of scorable traits corresponding to the number of components in the rubric, and the final predicted score for a response is the sum of the individual binary trait scores. Because we know exactly which rules, groups, evidence, and scorable traits were found or not found when scoring, we can automatically construct an explanation of our predicted scores. See Figure 1 for an example of a scorable trait and the score and explanation it produces.

## 5 Automated Discovery of Rules

Given the hierarchy of rules, groups, evidence, and scorable traits described above, one approach to developing a scoring model would be to define all of these elements manually. While manually constructed models perform well (per our experiments below), requiring manual effort to construct a scoring model prevents the adoption of this approach at any scale larger than a small handful of items. Thus, we would like to automate this process. However, our model is not differentiable, so approaches such as stochastic gradient descent can not be used.

Simulated annealing is a highly flexible opti-

mization technique that makes few assumptions about the objective function being optimized (Kirkpatrick et al., 1983). When applied to our modeling task, simulated annealing maximizes the performance of a model by iteratively adding or removing rules. If a change increases the model’s training set performance, we keep it. Otherwise, the change is stochastically accepted with a probability based on a temperature variable and the difference in performance between the new and previous states. As the procedure continues over many iterations, the temperature is slowly reduced according to a cooling schedule. The result of this is a process that initially makes many random changes, but that tends towards only making changes that maximize the performance of the model as the temperature decreases.

In practice, we evaluate the performance of our models using both accuracy and the unweighted average recall (UAR), and so we optimize against both of these metrics during the annealing process. That is, our goal is to maximize the following function:

$$S(\theta) = \lambda * \text{UAR}(\hat{y}_\theta) + (1 - \lambda) * \text{Acc}(\hat{y}_\theta)$$

where  $\theta$  corresponds to the model parameters, i.e., the rules, groups, and evidence of the model,  $\hat{y}_\theta$  to the predictions of the current model on the training set, and  $\lambda$  is a hyperparameter that controls the

relative importance of UAR versus accuracy.

To use simulated annealing, we must define the ways in which an existing model can be altered to generate a new model. We begin by building a set of candidate rules. Candidate math expressions are generated by identifying sequences of alternating operands and operators in the math extracted from a response. In this work, we consider sequences of up to six operands. Once these expressions have been identified, we rank them according to their information gain. We keep the top  $n$  expressions as our set of candidate rules for use in annealing.

When humans craft manual rules, they are able to write regular expressions. Automatically determining useful regular expressions in full generality is beyond the scope of this work, but providing our automated rules with some ability to reason about prose writing is important. For this reason, we consider all words in the responses, again rank by information gain, and then keep the top  $m$  as regular expression rules (that ultimately will match if the given word is present in the response).

When annealing our rules, we allow for four transformations:

1. Add a rule to a group.
2. Remove a rule from a group.
3. Replace a rule with a new rule.
4. Move a rule from one group to another group.

We initialize our model to have a number of scorable traits equal to the maximum score for the item, and create a user-defined number of empty evidences and groups for each trait. To improve final model performance, we use random restarts during training. That is, we perform  $k$  simulated annealing runs, and keep the model with the best training set performance as our final trained model.

To avoid overfitting to our training data, we also include two regularization terms in our objective function. The first term,  $R(\theta)$ , penalizes the model by the total number of operands used by all rules. The second term,  $E(\theta)$ , penalizes the model for the number of non-empty evidences used by the model. Our final objective function is

$$S'(\theta) = S(\theta) + \gamma * (\alpha * R(\theta) + \beta * E(\theta))$$

where  $\alpha$ ,  $\beta$ , and  $\gamma$  are hyperparameters that control the relative and overall regularization strength.

## 6 Experiments

To the best of our knowledge, there is no publicly available dataset that features open CR math word problems with a large number of student responses per item. For example, the GSM8k Dataset used in Table 1 has only one response per item. Therefore, we use our own proprietary dataset of MPT items for our experiments. This dataset consists of 14 items covering algebra, arithmetic, and geometry, targeting grade levels from fourth grade to high school. The scoring scales for these items range from 0–2 to 0–4. See Table 2 for detailed per-item information.

Our primary goal is to evaluate the performance of our rules-based model, both with manually crafted rules and automatically learned rules. The manual rules used in these experiments were crafted by human experts, who were allowed to view only a randomly sampled subset of the responses for each item. Responses used in this way during rule creation were also used for hyperparameter search for the simulated annealing approach, but were excluded from the dataset used in the final experiments. The response counts in Table 2 correspond to the counts used in our final experiments.

We perform a grid search for the cooling rate, number of iterations to run annealing for, and the overall regularization strength  $\gamma$ . Our pool of candidate rules consists of the top 500 expressions and top 50 words. We spend 1000 iterations at each temperature, create 3 positive evidences and 1 negative evidence for each trait, allow up to 10 groups per evidence, and set  $\alpha = 0.0025$ , and  $\beta = 0.01$ . We use a geometric cooling schedule, and perform 5 random restarts. These settings are based on values that were found to work well during initial development. We use 5 stratified and randomized train/test splits when performing this hyperparameter search, with 25% of the data in the test split.

Prior work has found that traditional AES approaches can work well for MPT, such as random forests (Erickson et al., 2020) and recurrent neural networks (Cahill et al., 2020). For this reason, we compare our rules-based scoring to three other conventional approaches: fine-tuned DistilBERT (Sanh et al., 2019), character n-gram random forests, and word n-gram random forests.

For both random forest models, we use regression random forests with 100 trees, and 33% of the features considered at each split. We keep all n-grams that occur in more than 5% of documents

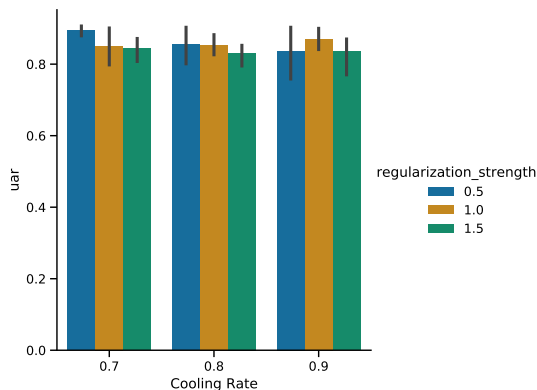


Figure 2: Mean UAR achieved by annealing on Item 6 at various regularization strengths and cooling rates at 25,000 annealing iterations. We also evaluated performance at 50,000 and 75,000 iterations, but performance across all three settings was similar, so we show only the results for 25,000 for clarity. Error bars are 95% bootstrap confidence intervals.

and in fewer than 95% of documents. For character n-grams, we consider n-grams ranging from 3 to 6 characters long. For word n-grams, we consider n-grams ranging from 1 to 4 words. We use scikit-learn’s implementations of random forests and count vectorizers (Pedregosa et al., 2011).

For the DistilBERT model, we finetune all layers using the Adam optimizer. We use a learning rate of  $2e-5$ , a weight decay of 0.01, and train for 4 epochs. The training data is further split into a final training set and an evaluation set; we evaluate model performance on the evaluation set after each epoch, and we evaluate our final test-set performance on the model that achieved the best evaluation set performance. DistilBERT uses wordpiece tokens (Wu et al., 2016) with a 512 token context window. All of our responses fit within this window; the longest response in our dataset is 501 tokens long. Our DistilBERT fine-tuning utilizes Hugging Face (Wolf et al., 2020).

For both random forests and DistilBERT finetuning, all hyperparameters not mentioned here were left at their default values.

For each item, we create 30 stratified and randomized train/test splits, with 25% of the data in the test split, and train and evaluate all models on these splits. We evaluate model performance using both accuracy and the unweighted average recall (UAR). In our operational scoring, poor performance at any scorepoint can rule out the use of a model, and UAR captures this by considering the impact of poor performance at rare and common scorepoints

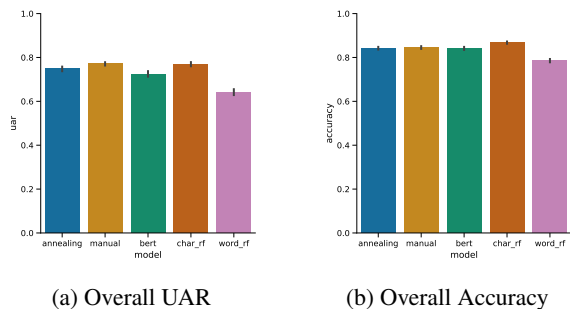


Figure 3: Mean UAR and accuracy of each approach, averaging over all items and folds. Error bars are 95% bootstrap confidence intervals.

to be equivalent. For all regression models, we generate final score predictions by rounding the model output to the nearest whole number.

## 7 Results and Discussion

The results of our hyperparameter grid search for simulated annealing are shown in Figure 2. We see that performance is quite robust across all hyperparameter settings tested. Best performance is achieved by annealing for 25,000 iterations, with a cooling rate of 0.7 and a regularization strength of 0.5. These are the settings that we use for simulated annealing in the other experiments described in this section.

The mean UAR and accuracy of each model, averaging over all items and folds, is shown in Figure 3. Focusing on UAR, we see that the random forest using word n-grams performs noticeably worse than the other approaches. Character n-gram random forests and manually crafted rules perform well. Finally, we see that our annealing-based approach to automatically constructing rules performs slightly worse than the manually crafted rules, but slightly better than the DistilBERT model.

When we compare accuracy trends, we see that our rules-based approaches perform no better than DistilBERT. This is due to performance at the lowest scorepoints - these tend to be common (and thus prominent in the calculation of accuracy), but the rules-based approaches tend to have slightly lower recall at the lowest scorepoint. This is not seen in the UAR figures because the rules-based models tend to perform slightly better on the higher (and rarer) scorepoints.

In Figure 4, we show the mean UAR of each model for all items. Our discussion here will focus on items 2 and 10; these items were chosen as examples where the annealing approach performs

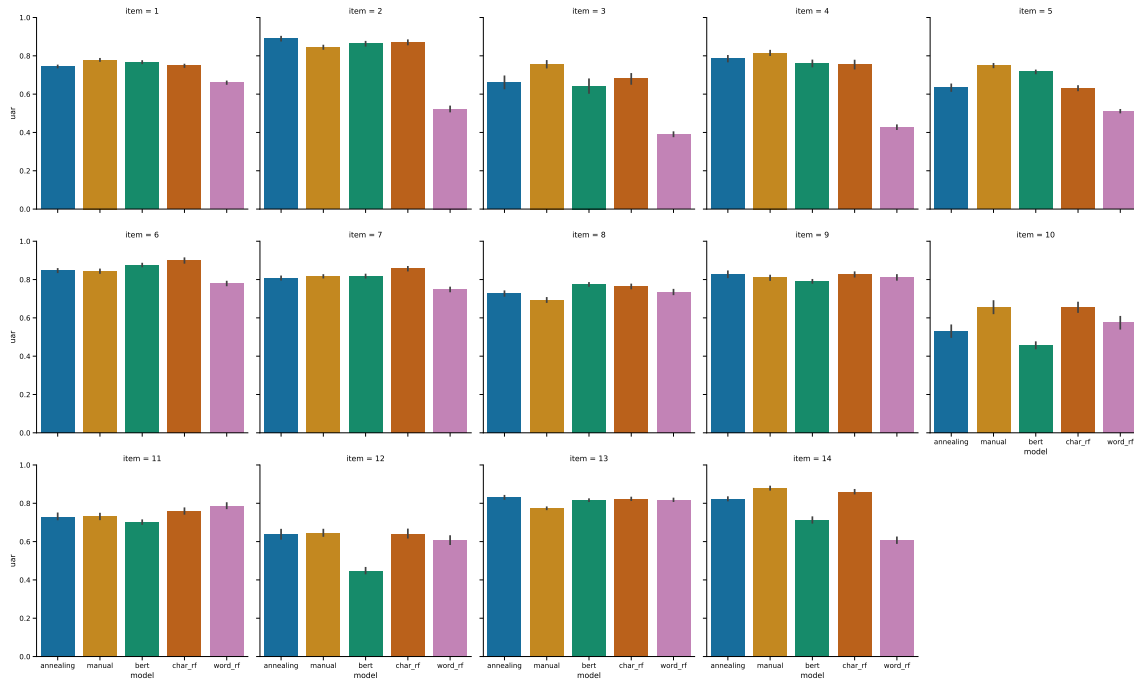


Figure 4: Mean UAR per item, averaging over all folds. Error bars are 95% bootstrap confidence intervals.

very well and very poorly, respectively.

For Item 2, our annealing approach performs the best out of all models. This item describes the improvement in average speed of two athletes over the course of a training regimen, and asks the students to calculate at what week of training their average speeds will be equal. The rubric contains a computation component, requiring the students to calculate the correct week, and a modeling component, requiring students to show their work in calculating their answer. The annealing process successfully constructs evidence both for identifying when the correct answer is present, and for identifying work that supports that correct answer.

In contrast, for Item 10, the annealing process performs quite badly. Item 10 asks students to calculate the speed of a real car based on the performance of a scale model of that car. The rubric contains one computation component, for the correct final speed, as well as two modeling components, one for proper unit conversion and one for correctly scaling the speed to the full-size car. The manually crafted rules perform comparably to the character n-gram random forest for Item 10, indicating that it is possible for our rules-based approach to perform relatively well on this item. However, our manual rules for this item make extensive use of regular expressions, both to capture information about units and to capture notions such as the student stating

in prose that they multiplied by the scaling factor. These sorts of sophisticated regular expressions are not captured by our current candidate rule generation process.

The relatively lackluster performance of the DistilBERT model is surprising, given the dominance of transformer-based approaches in many areas of NLP. However, there is a substantial literature detailing how both recurrent and transformer-based neural models can struggle with mathematics (Huang et al., 2018; Cobbe et al., 2021; Hendrycks et al., 2021). This literature, in combination with our results here, suggests that fine-tuning off-the-shelf neural models is not a particularly powerful approach for MPT scoring.

In light of these results, we conclude that our rules-based approach enables explainable automated scoring of MPT items without sacrificing performance, at the cost of requiring manual effort in designing the rules. However, we also have found that a simulated annealing-based approach to automatic rule creation can produce explainable models that are almost as effective as manually crafted rules, allowing for scalable and explainable MPT scoring.

## 8 Conclusion and Future Work

We have presented a novel, explainable approach to scoring MPT items via handcrafted rules that



performs well, and have shown that such rules can be automatically discovered through simulated annealing.

While our model is able to provide explanations of its scores, generating explanations is only the first step in the full explainability process. Explanations are of limited utility without the ability to convey model explanations to stakeholders such as test takers or test administrators. Determining how best to use the explanations produced by our models is an important area of future work.

Our approach is heavily reliant on the assumption that the final score of a response is the sum of multiple binary components. For MPT items that are not structured in this way, it is unlikely that our approach would work well on its own, although it could possibly be combined with other approaches. We are actively investigating how best to extend our approach to more rubric types.

The success of our annealing process ultimately relies on our ability to generate useful candidate rules. While our current process works well, we have seen that for some items, we need to be able to construct more sophisticated rules. Determining how to improve the generation of our candidate pool is another promising area for future work.

The dataset we used in this work is mainly composed of algebra problems. While we do have some geometry and arithmetic items, how well our approach can generalize to other MPT item types is an area of future work. In particular, our items do not cover calculus, trigonometry, or other areas that require students to extensively reason about functions.

## Acknowledgements

We would like to thank Alicia Bouy for her assistance in constructing the manually-crafted rules, and Lee Becker and Joshua Southerland for their feedback during the writing process.

## References

- Sami Baral, Anthony F Botelho, and John A Erickson. 2021. Improving Automated Scoring of Student Open Responses in Mathematics. In *Proceedings of The 14th International Conference on Educational Data Mining (EDM21)*, page 9, Paris, France.
- Philip G. Butcher and Sally E. Jordan. 2010. A comparison of human and computer marking of short free-text student responses. *Computers & Education*, 55(2):489–499.

Aoife Cahill, James H Fife, Brian Riordan, Avijit Vajpayee, and Dmytro Galochkin. 2020. [Context-based Automated Scoring of Complex Mathematical Responses](#). In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 186–192, Seattle, WA, USA → Online. Association for Computational Linguistics.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Denise Dellarosa Cummins, Walter Kintsch, Kurt Reusser, and Rhonda Weimer. 1988. [The role of understanding in solving word problems](#). *Cognitive Psychology*, 20(4):405–438.

Fei Dong, Yue Zhang, and Jie Yang. 2017. [Attention-based Recurrent Convolutional Neural Network for Automatic Essay Scoring](#). In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 153–162, Vancouver, Canada. Association for Computational Linguistics.

John A Erickson, Anthony F Botelho, Steven McAteer, Ashvini Varatharaj, and Neil T Heffernan. 2020. The automated grading of student open responses in mathematics. In *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*, pages 615–624.

James H Fife. 2017. The m-rater™ Engine: Introduction to the Automated Scoring of Mathematics Items. Technical Report ETS RM–17-02.

Scott Hellman, Mark Rosenstein, Andrew Gorman, William Murray, Lee Becker, Alok Baikadi, Jill Budden, and Peter W. Foltz. 2019. [Scaling Up Writing in the Curriculum: Batch Mode Active Learning for Automated Essay Scoring](#). In *Proceedings of the Sixth (2019) ACM Conference on Learning @ Scale, L@S '19*, pages 1–10, New York, NY, USA. Association for Computing Machinery.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

Danqing Huang, Jing Liu, Chin-Yew Lin, and Jian Yin. 2018. [Neural math word problem solver with reinforcement learning](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 213–223, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

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, Berlin, Germany. Association for Computational Linguistics.

- J. C. S. Kadupitiya, Surangika Ranathunga, and Gihan Dias. 2017. [Assessment and Error Identification of Answers to Mathematical Word Problems](#). pages 55–59. ISSN: 2161-377X.
- Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. 1983. Optimization by simulated annealing. *science*, 220(4598):671–680.
- Sachin Kumar, Soumen Chakrabarti, and Shourya Roy. 2017. [Earth Mover’s Distance Pooling over Siamese LSTMs for Automatic Short Answer Grading](#). Pages: 2052.
- Nate Kushman, Yoav Artzi, Luke Zettlemoyer, and Regina Barzilay. 2014. [Learning to Automatically Solve Algebra Word Problems](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 271–281, Baltimore, Maryland. Association for Computational Linguistics.
- Andrew S. Lan, Divyanshu Vats, Andrew E. Waters, and Richard G. Baraniuk. 2015. [Mathematical Language Processing: Automatic Grading and Feedback for Open Response Mathematical Questions](#). In *Proceedings of the Second (2015) ACM Conference on Learning @ Scale, L@S ’15*, pages 167–176, New York, NY, USA. Association for Computing Machinery.
- Leah S. Larkey. 1998. [Automatic essay grading using text categorization techniques](#). In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR ’98*, pages 90–95, Melbourne, Australia. ACM Press.
- Claudia Leacock and Martin Chodorow. 2003. [C-rater: Automated Scoring of Short-Answer Questions](#). *Computers and the Humanities*, 37(4):389–405.
- Nava L Livne, Oren E Livne, and Charles A Wight. 2007. Can Automated Scoring Surpass Hand Grading of Students’ Constructed Responses and Error Patterns in Mathematics? *MERLOT Journal of Online Learning and Teaching*, 3(3):12.
- Ellis B Page. 1966. The imminence of grading essays by computer. *The Phi Delta Kappan*, 47(5):238–243.
- 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(Oct):2825–2830.
- Lakshmi Ramachandran, Jian Cheng, and Peter Foltz. 2015. [Identifying Patterns For Short Answer Scoring Using Graph-based Lexico-Semantic Text Matching](#). In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 97–106, Denver, Colorado. Association for Computational Linguistics.
- Brian Riordan, Andrea Horbach, Aoife Cahill, Torsten Zesch, and Chong Min Lee. 2017. [Investigating neural architectures for short answer scoring](#). In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 159–168, Copenhagen, Denmark. Association for Computational Linguistics.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *5th Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS 2019*, volume abs/1910.01108.
- Jia Tracy Shen, Michiharu Yamashita, Ethan Prihar, Neil Heffernan, Xintao Wu, Ben Graff, and Dongwon Lee. 2021. [MathBERT: A Pre-trained Language Model for General NLP Tasks in Mathematics Education](#). In *Math AI For Education Workshop*.
- Mark D. Shermis and Jill C. Burstein, editors. 2003. *Automated essay scoring: A cross-disciplinary perspective*. Lawrence Erlbaum Associates, Inc., Mahway, NJ.
- Mark D. Shermis and Jill C. Burstein, editors. 2013. *Handbook of automated essay evaluation: Current applications and new directions*. Routledge, New York.
- Nina V Stankous. 2016. Constructive response vs. multiple-choice tests in math: American experience and discussion. In *2nd PAN-AMERICAN INTER-DISCIPLINARY CONFERENCE, PIC 2016 24-26 February, Buenos Aires Argentina*, page 321.
- Lynn Streeter, Jared Bernstein, Peter Foltz, and Donald DeLand. 2011. [Pearson’s Automated Scoring of Writing, Speaking, and Mathematics](#). Technical report.
- Kaveh Taghipour and Hwee Tou Ng. 2016. [A Neural Approach to Automated Essay Scoring](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1891, Austin, Texas. Association for Computational Linguistics.
- Luis Tandalla. 2012. [Scoring Short Answer Essays](#). Technical report.
- Yan Wang, Xiaojiang Liu, and Shuming Shi. 2017. Deep neural solver for math word problems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 845–854.
- 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.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. [Google’s neural machine translation system: Bridging the gap between human and machine translation](#). *CoRR*, abs/1609.08144.

Zhipeng Xie and Shichao Sun. 2019. [A Goal-Driven Tree-Structured Neural Model for Math Word Problems](#). In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 5299–5305, Macao, China. International Joint Conferences on Artificial Intelligence Organization.

# Gender-Inclusive Grammatical Error Correction through Augmentation

Gunnar Lund, Kostiantyn Omelianchuk, Igor Samokhin

Grammarly

{gunnar.lund,kostiantyn.omelianchuk,igor.samokhin}@grammarly.com

## Abstract

In this paper we show that GEC systems display gender bias related to the use of masculine and feminine terms and the gender-neutral singular *they*. We develop parallel datasets of texts with masculine and feminine terms, and singular *they*, and use them to quantify gender bias in three competitive GEC systems. We contribute a novel data augmentation technique for singular *they* leveraging linguistic insights about its distribution relative to plural *they*. We demonstrate that both this data augmentation technique and a refinement of a similar augmentation technique for masculine and feminine terms can generate training data that reduces bias in GEC systems, especially with respect to singular *they* while maintaining the same level of quality.

## 1 Introduction

Natural Language Processing (NLP) systems are well known to exhibit sensitivity to social characteristics, a sensitivity that may lead to harms for users interacting with these systems. In this work, we examine how NLP systems performing the task of Grammatical Error Correction (GEC) are sensitive to *gender* characteristics in English and how this sensitivity represents bias that harms users of these systems. We propose a new data augmentation technique to address bias related to singular *they*, and show how it can mitigate gender bias. We also show how this technique interacts with prior work on data augmentation methods to reduce bias.

The expression of gender in English is complex, and to focus our study, we identify discrete biases that GEC systems may exhibit. First, we build on other works on gender bias in NLP by examining how masculine and feminine terms impact the behavior of GEC systems. Discrepant behavior between sentences that contain one or the other is evidence of bias. Second, we examine the behavior of GEC systems with a gender-neutral pronominal

paradigm in English commonly called singular *they*. The linguistic properties of this paradigm introduce additional kinds of biases relative to masculine or feminine pronouns. Additionally, we make our evaluation datasets publicly available<sup>1</sup>.

To prevent these adverse behaviors, we apply techniques to generate synthetic training data that address different aspects of these behaviors. First, we adopt Counterfactual Data Augmentation (CDA), which is used successfully to reduce bias in word embedding models (Lu et al., 2019; Maudslay et al., 2020), and apply it to the GEC case. Second, we introduce a new technique for generating training data with *they* pronouns that have unambiguously singular reference using insight from theoretical linguistics to target model bias with gender-neutral sentences. Because these techniques are data-oriented, the innovations should theoretically generalize beyond the GEC domain to other NLP tasks as well.

Our main contributions are:

- We introduce a novel technique for creating singular *they* data, leveraging specific linguistic features of this use of the pronoun. Additionally, we refine previous approaches to addressing discrepancies in model behavior between texts with masculine and feminine pronouns and show their application outside of the context of masked language models.
- We qualitatively and quantitatively measure biases in competitive GEC systems by comparing model performance on parallel test sets containing singular *they* pronouns, masculine terms, and feminine terms.
- We show how our data augmentation techniques, both in isolation and combination, mitigate these biases when used to create training

<sup>1</sup><https://github.com/grammarly/gender-inclusive-gec>

data for these GEC systems with a minimal impact to overall performance.

## 2 Background

### 2.1 Gender and Bias in NLP and GEC

#### 2.1.1 Conceptual grounding

To orient this work, we highlight two recent calls-to-action regarding the study of bias and gender in NLP systems. First, following [Devinney et al. \(2022\)](#), we mean to be explicit about the conception of gender and gendered language we assume. In particular, we are concerned with gendered linguistic *content*, and not the gender of the authors or readers of that content. We recognize that the expression of gender in English is notional; the use of some nouns and pronouns is linked to particular gendered conceptual categories ([McConnell-Ginet, 2013](#); [Ackerman, 2019](#)). Additionally, the *use* of language with gendered content represents one aspect of gender performativity which produces and reifies these gendered categories.

Second, following [Blodgett et al. \(2020\)](#), “bias” is an inherently normative concept. In the context of NLP systems, it must be understood in terms of the potential *harms* that those systems may cause and to *whom* those harms may be caused. Therefore, we directly focus on mitigating harms themselves as they relate to the GEC task and how users of these systems interact with them and may be affected by them. Unlike some studies of bias in upstream contexts like word embeddings, users interact with GEC systems directly; these users choose to incorporate these systems’ suggestions into their emails, essays, and tweets, and bias may impact anyone interacting with such text.

Further, because the GEC task is an inherently normative one on its own—these systems offer suggestions to *correct* a user’s text and are designed in accordance with preexisting normative notions of “correct” or “fluent” English—GEC systems necessarily also participate in the production of gendered categories. The norms assumed when constructing these systems and datasets in this regard may conflict with other norms about language use. For example, there are norms against the use of singular *they* in some language communities. Some English speakers do not accept singular *they* as a grammatical construction of English ([Bjorkman, 2017](#), a.o.), and some prescriptive grammars advise against the use of singular *they* (c.f., [Strunk and White, 1999](#)). People who are non-binary and use *they* pronouns,

e.g., cannot refer to themselves “correctly” within these circumscribed norms of language use. As we discuss below, the operationalization of these norms in GEC systems may lead to harm. We adopt the view that GEC systems should reflect the most permissive distribution of singular *they*. This distribution is discussed further in section 3.2.

Our work has a notable limitation in that we do not investigate bias with respect to neopronouns like *ze* or *xe*. We leave extensions of CDA-like techniques to these pronouns for future work.

#### 2.1.2 Two biases

We identify two areas where GEC systems can produce biased, and therefore potentially harmful, outcomes. Importantly, this is not an exhaustive account of potential biases GEC systems exhibit, but we think this is as good a starting point as any.

First, a GEC system can be implicitly biased if it consistently performs better on texts containing words of one gendered category over another. This is an allocative harm. If a GEC system performs worse on texts that are about people who use one pronoun or another, texts about those people may contain more grammatical errors, impacting their relative opportunity. For example, a user writing letters of recommendation may inadvertently include more grammatical errors in letters for individuals using masculine pronouns, as a system could perform worse on texts with masculine pronouns than feminine pronouns, and this could impact the relative reception those letters receive compared to similar letters with feminine pronouns.

Second, a GEC system can be explicitly biased if it offers corrections that reify harmful notions about particular gendered categories, including the reinforcement of stereotypes and misgendering or erasure of individuals referred to in the user’s text. This is a representational harm. This harm is explicitly called out by participants in a survey on harms of AI systems with respect to non-binary individuals ([Dev et al., 2021](#)). The examples below are representative of these kinds of corrections. The first is an instance of misgendering, replacing singular *they* with a masculine pronoun; the second is an instance of erasure, implying that *they* has a correct use only as a plural pronoun.

1. I asked Alex **their** phone number. -> I asked Alex **his** phone number.
2. They are **a linguist**. -> They are **linguists**.

We find evidence for both of these biases by analyzing the following GEC systems (table 1)

1. GECToR (Omelianchuk et al., 2020), sequence tagging approach, which was a state-of-the-art GEC model in 2020
2. Fine-tuned BART model (Lewis et al., 2020), which represents another popular and competitive approach - sequence to sequence
3. EditScorer (Sorokin, 2022), the recent ranker approach, that is the second-best result on BEA Shared Task 2019, as of April 2023<sup>2</sup>.

Our quantitative analysis revealed that, for all three systems, there is a significant gap (from -6.2% to -9.5% F05 points) between the original and augmented with singular *they* examples versions of the BEA-dev subset, which we call bea-195. The details on how we built these datasets and evaluation approach is provided in sections 4.1 and 4.2. Detailed evaluation results are available in appendix table 7.

We hypothesize that these biases share a partial cause: an imbalance in the training data. If, e.g., the training data with masculine words is of a higher quality than that with feminine words, there may be a performance gap. In the case of unnecessary corrections of singular *they*, we hypothesize that the imbalance is caused by an extreme lack of singular *they* sentences relative to plural *they* sentences. In the remainder of this paper, we show that the introduction of synthetic data helps mitigate these biases.

## 2.2 Related work

### 2.2.1 Data augmentation

Data augmentation has been used in other NLP domains to mitigate gender bias, but most of these works focus on just the masculine and feminine gender categories in English and limit the application of these techniques to word embedding models. Zhao et al. (2018); Rudinger et al. (2018); Lu et al. (2019) show that coreference resolution systems are sensitive to masculine and feminine words in otherwise equivalent sentences. Lu et al. (2019) use what they call Counterfactual Data Augmentation (CDA) to reduce this sensitivity. In CDA, masculine pronouns are swapped for feminine ones

and vice versa. They also swap definitionally gendered common nouns like *actor* and *actress*. They set aside data where the swapping candidates are in a cluster with a proper name.

Maudslay et al. (2020) extend Lu et al. (2019) CDA and implement additional name swapping, where the gendered associations of names were determined using census data from the US Social Security Administration. They use this technique to minimize gendered differences in word embedding spaces as measured by WED (Bolukbasi et al., 2016). As in Lu et al., Maudslay et al. limit their method to masculine and feminine categories.

In addition, gender-neutral data augmentation methods have been proposed in concurrent works by Sun et al. (2021) and Vanmassenhove et al. (2021). These methods have different goals than ours and are designed to produce different kinds of data. We discuss the differences between these methods and our own in section 3.3.

To our knowledge, ours is the first work to use CDA techniques to reduce bias in GEC systems and the first to use singular *they* augmentation to inject synthetic training data to reduce bias with singular *they* sentences.

### 2.2.2 Singular *they* and NLP systems

Previous works investigating bias towards singular *they* sentences have generally focused on coreference resolution systems. Cao and Daumé III (2021) develop a dataset to evaluate these systems on naturalistic texts about individuals who identify as non-binary, where 35% of the pronouns are singular *they*. They report that the Stanford system is the highest scoring on this dataset with an F1 score of 34.3%. This same system reports a much higher F1 score of 60% on the CONLL 2012 test set.

Baumler and Rudinger (2022) compare coreference resolution system performance directly on singular *they* sentences compared to plural *they* sentences along the lines of the Winograd or Winogender schemata (Levesque et al., 2012; Rudinger et al., 2018; Zhao et al., 2018). They find across-the-board gaps in system performance between the two test sets.

Outside of coreference resolution, Dev et al. (2021) investigate biased representations with BERT in a masked word prediction task. They find that for masked pronouns, BERT has a high accuracy in the prediction of masculine and feminine pronouns, but accuracy considerably lowers for singular *they*.

<sup>2</sup>[http://nlpprogress.com/english/grammatical\\_error\\_correction.html](http://nlpprogress.com/english/grammatical_error_correction.html)

System	bea-dev-full	bea-195			bea-556		
	F05	F05 orig	F05 st aug	diff	F05 orig	F05 mf aug	diff
GECToR (roberta-base)	54.57%	58.28%	48.74%	-9.54%	59.23%	58.96%	-0.27%
BART (seq2seq)	52.74%	56.36%	50.13%	-6.23%	58.61%	58.79%	0.18%
EditScorer (roberta-large)	58.92%	60.6%	54.16%	-6.44%	62.55%	61.59%	-0.96%

Table 1: Scores on BEA-dev subsets for strong GEC baselines.

### 3 Description of the data augmentation methods

We use two data augmentation methods. First, we follow Lu et al. (2019) and others in swapping out feminine words for masculine words and vice versa. Second, we propose a novel augmentation method for generating singular *they* data from sentences containing masculine and feminine pronouns. We treat singular *they* differently because language internal facts about English necessitate a separate treatment: unlike *he* and *she*, *they* has a second life as a plural pronoun.

#### 3.1 Feminine/Masculine CDA (FM-CDA)

Consistent with masculine/feminine-term swapping methods in other works, we swap three kinds of nominal terms:

- Pronouns: Swap masculine pronouns for their feminine counterparts and vice versa. Ex: *him* → *her*. Because the masculine and feminine pronominal paradigms are partly syncretic—the feminine pronoun *her* can be accusative or possessive and map to *him* or *his*, respectively—token POS tags, generated by a proprietary POS tagger, were used to appropriately match terms to their case-same counterpart.
- Common nouns: Swap definitionally feminine common nouns for their masculine gendered counterparts and vice versa. Ex: *actor* → *actress*. The selection and mapping of these nouns were hand-curated by industry experts.
- Names: Swap first names that are usually associated with feminine terms for names usually associated with masculine terms and vice versa. We partnered with industry experts to curate dictionaries of masculine and feminine names. Because names don’t necessarily have gendered counterparts in the way that pronouns or common nouns do, an arbitrary mapping of names was created. Names occurring in both lists were excluded from swapping.

Unlike some previous work involving CDA on fully unsupervised tasks where the creation of a single counterpart sentence is sufficient, GEC training data consists of pairs of ungrammatical source text and grammatical target text. This introduces challenges similar to those that CDA faces for machine translation data consisting of parallel texts (Saunders and Byrne, 2020; Wang et al., 2022) Because, e.g., the POS tagger may perform differently on the two texts, especially given that the source text is ungrammatical, the swapping algorithm may produce inconsistent swaps if applied separately to the source and target texts. This inconsistency can introduce grammatical errors between the source and target texts and negatively impact model performance.

To avoid this, an additional algorithm ensures a consistent swap between the source and target text where possible. We first apply our algorithm to the grammatically corrected target text. Then we use an alignment algorithm to align the target and source texts and isolate the differing segments of the texts. For each differing segment, we determine if the number of tokens in the source and target segments is the same, and if not, we discard the data. Then we compare every token in the source and target segments; if the source word would have the same swap as the target, we replace the source word with its differently gendered counterpart. If neither word is swappable, we do nothing. If there is a mismatch in the swap between source word and target word, we discard the data point. For the evaluation set discussed below, we reintroduced this discarded data and manually edited the data to introduce singular *they* pronouns and ensure that the results are parallel.

#### 3.2 Singular *they* CDA (St-CDA)

Singular *they* is an inherently referential phenomenon. As is evident in the name, it is distinguished from plural *they* because it refers to singular individuals. Further, theoretical and experimental linguistic works show that the overall distribution of singular *they* is conditioned by the nature of

the singular antecedent and discourse participants’ relation to the antecedent (Bjorkman, 2017; Ackerman, 2019; Moulton et al., 2020; Konnelly and Cowper, 2020; Han and Moulton, 2022). Speakers may be sensitive to linguistic vs. non-linguistic antecedents (Moulton et al., 2020), specificity and definiteness of the antecedent (Bjorkman, 2017; Konnelly and Cowper, 2020), the discourse participant’s knowledge of the referent’s gender identity (Bjorkman, 2017; Ackerman, 2019; Konnelly and Cowper, 2020), and the association of a lexical item or name with a particular gender category (Bjorkman, 2017; Ackerman, 2019; Moulton et al., 2020). In the case of the broadest distribution, singular *they* is used in the same ways that masculine and feminine pronouns are used—the referent’s preference largely dictates the choice of pronoun—but *they* may additionally be used when the referent’s preference is not known (Konnelly and Cowper, 2020).

Differently than feminine and masculine pronouns, we hypothesize that adverse model behavior with singular *they* is at least partially caused by its infrequency relative to plural *they*. Therefore, it is not enough to simply create data that has *they* pronouns; it must also be evidently singular as well. We leverage these linguistic insights about antecedenthood to identify contexts where swapping will result in unambiguous cases of singular *they* by identifying singular antecedents of the pronouns in the text.

We implement this by using HuggingFace’s Neuralcoref coreference resolution system<sup>3</sup> built on top of SpaCy<sup>4</sup>. For a given coreference cluster with a masculine or feminine pronoun, we look at the coreferring expressions in the cluster. If we find a singular one, we perform the swap. We consider a coreferring expression singular if it is:

1. A singular common or proper noun.
2. A singular possessum (e.g., *his foot*).

In addition, *they* has different verbal agreement paradigms than *he* or *she*. We resolve this by using SpaCy’s dependency parser to identify agreeing verbs with the swapped pronouns. We then use the pyInflect package<sup>5</sup> to select the verbal inflection consistent with subject agreement with *they*.

<sup>3</sup><https://github.com/huggingface/neuralcoref>

<sup>4</sup>Neuralcoref works with SpaCy v2.1 (<https://v2.spacy.io/>).

<sup>5</sup><https://github.com/bjascob/pyInflect>

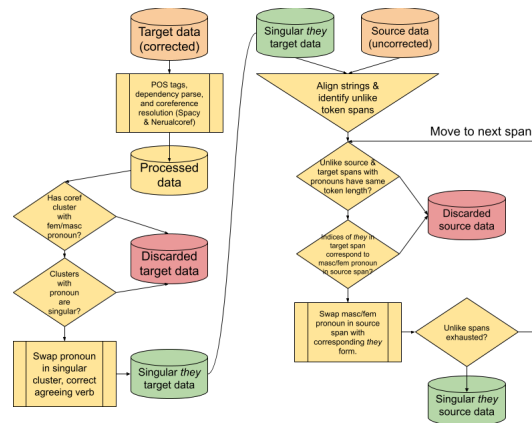


Figure 1: A graphical illustration of the St-CDA swapping process.

Finally, we use POS information to disambiguate syncretic forms of *her*. For reflexive pronouns, we swap in the form *themselves* and not *themselves* as this form is much less common in the preexisting training data and is more likely to lead to a singular interpretation of *they* pronouns.

As in the case of FM-CDA, we face the potential for inconsistency if this technique is used separately for both source and target data. We use the same algorithm as FM-CDA to perform safe swaps with an additional check on the verbs that were corrected to agree with *they* in the target swapped data.

### 3.3 Differences between St-CDA and other approaches

To our knowledge, there are two similar approaches to generating singular *they* data in (Sun et al., 2021) and (Vanmassenhove et al., 2021). Their approaches differ from ours in two crucial ways. First, their techniques are meant to create wholly gender-neutral texts, swapping out instances of definitionally gendered noun phrases like “fireman” for “firefighter”. We do not perform these swaps because *they* pronouns may corefer with definitionally gendered words, and this data, in particular, is likely to be rare in most corpora. As we see qualitatively, the baseline model we test seems to be particularly likely to “correct” singular *they* sentences unnecessarily when there is a definitionally gendered word in the sentence.

Second, and most crucially, we hypothesize that the performance gap with singular *they* sentences is due to the relative glut of plural *they* data compared to singular *they* data. As such, we seek to add *they* data that unambiguously has singular reference. Their techniques, by contrast, may result in



data where *they* may have a primarily plural interpretation. By targeting contexts where *they* is more likely to be interpreted singularly, we believe St-CDA produces data that will have a higher positive impact and have fewer adverse effects on model quality.

Ultimately, Sun et al. (2021) and Vanmassenhove et al. (2021) have different goals than we do, and this informs the differences in our techniques. Both works envision their technique to be used at runtime in machine translation tasks to, e.g., ensure translations from languages with grammatical gender result in gender neutral translations in English. While they speculate that their techniques can be used to create augmented training data, as we do in this paper, they do not specify what issues they intend this augmented training data to address. By contrast, we seek to counteract a particular imbalance between plural and singular *they* sentences.

## 4 Experiments with GEC

### 4.1 Description of datasets

For training data, we chose a large Lang-8 Corpus of Learner English (Mizumoto et al., 2011), and more specifically, its “cleaned” version cLang-8 (Rothe et al., 2021) which contains over 2 million corrected English sentences. The downside is the noisiness of the data (even in the “cleaned” version) and the lack of consistency in annotations. There are few other GEC datasets of comparable size (Bryant et al., 2022). Naturally, not every sentence contains personal pronouns, so only a subset of the dataset is suitable for data augmentation. The size of cLang-8 allowed us to produce about 63 thousand sentences with singular-they augmentation and 254 thousand gender-swapped sentences, which is enough for fine-tuning purposes. In our further experiments, we used only a random sample of 50 thousand sentences from each augmented version of the data to make sure that the results were not impacted by a difference in data size.

### 4.2 Evaluation approach

#### 4.2.1 Description of the evaluation procedure

There is no evaluation set which would specifically contain multiple uses of the singular *they*, so we need to apply data augmentation here as well. To do this, we use the dev part of the BEA-2019 shared task (Bryant et al., 2019) since it is one of the standard evaluation sets for GEC. Of 4384 sentences in the BEA dataset, 195 singular *they*

sentences were created by replacing the pronouns “he” and “she” with singular “they.” To do this, we applied the CDA-st algorithm described above. The data discarded by the alignment algorithm was also collected and manually revised where possible (sentences where, e.g., a pronoun was inserted or changed from one gendered pronoun to another were either eliminated or revised to eliminate the error). Finally, the entire dataset was manually reviewed to ensure consistency between the original data and the augmented data.

To find the difference in GEC performance on sentences with and without the singular “they,” we evaluate on the subset of 195 sentences before augmentation, “BEA-195-orig”, and on the 195 augmented sentences, “BEA-195-st-aug”. The dataset size limits the conclusions we can make about the GEC model’s performance in general, but the differences between scores obtained by the GEC models on these two subsets are statistically significant.

We repeat this procedure for experiments involving masculine and feminine swapping. In this case, our augmentation produced subsets of 556 sentences: “BEA-556-orig” and “BEA-556-mf-aug”.

#### 4.2.2 Error distribution analysis

To ensure that our augmentation did not affect edits and shift the error distribution, we conducted a qualitative analysis of m2 files produced by Errant tool on parallel sentences of the original and augmented versions of “195” and “556” evalsets. As shown in the edit type distribution (appendix section B.1), there are only minor differences in the number of edits (less than 1% of edits affected). It can be explained by the fact that sometimes Errant might represent similar edits by single or multiple edits, like in the following example 2.

The error type distribution for both subsets is available in appendix section B.2.

#### 4.2.3 Questions to answer with evaluation

Running evaluation on these datasets, we are interested mainly in answering two questions:

- Is the state-of-the-art GEC model, which was not trained specifically with singular “they” or gender-swapped data, producing worse corrections on the augmented evaluation dataset?
- If the corrections are worse, can we shrink or remove the gap in performance by fine-tuning the model on the augmented training data?

Data source	Sentence	Edits
original bea-dev	I love this game because my favourite sport <b>man</b> belong to this game .	sport man belong to this => sportsman plays
mf aug bea-dev	I love this game because my favourite sport <b>woman</b> belong to this game .	sport woman => sportswoman belong to this => plays

Table 2: An example of a sentence with a different number of edits in an m2 file depending on data augmentation.

### 4.3 Description of models

For experiments, we use GECToR (Omelianchuk et al., 2021) - a state-of-the-art GEC model based on the efficient sequence tagging approach to corrections. Instead of producing a new error-free sentence, GECToR predicts a sequence of tags denoting operations: “keep,” “remove,” “insert\_X,” or “append\_X.” The corrected text is reconstructed from the original sentence and the tags. Sequence tagging is computationally cheaper than autoregressive approaches, which makes GECToR up to ten times faster than sequence-to-sequence models. At the same time, GECToR set the state-of-the-art at the time of publication.

GECToR is trained and fine-tuned in several stages, starting from the pre-trained language model such as RoBERTa (Liu et al., 2019). One can also start from the fine-tuned GECToR checkpoint (available on GitHub) and fine-tune it further on the data specifically tailored to the task at hand. However, it may lead the catastrophic forgetting issue, and the overall performance of the model on the general GEC test sets may deteriorate.

### 4.4 Experiment approach

We select GECToR for our fine-tuning experiments due to it being a competitive GEC system and having code that is publicly available. We use weights of the pre-trained GECToR (with RoBERTa-base encoder) model as initialization and fine-tune it for 5 epochs on the following data:

1. Original clang8 sentences ( 2.2m sentences)
2. Mix of original and augmented clang8 sentences of one type ( 2.2m + 50k sentences, either singular-they or gender-swapped)
3. Mix of original and augmented clang8 sentences of both types ( 2.2m + 100k sentences, 50k for both singular-they and gender-swapped)

We fine-tune the model for 5 epochs with early stopping after 3 epochs and 1 cold epoch. For

each training data configuration, we run training 10 times with different random seeds and report the average across all run results. The full list of hyperparameters for fine-tuning can be found in Appendix B.

Because the baseline GECToR model is already strong enough (it was a SOTA model in 2020) and clang-8 is high-quality data produced by another strong GEC system gT5 xxl (Rothe et al., 2021), the fine-tuning does not lead to substantial quality degradation. As shown in table 3, the differences in F0.5 scores are statistically insignificant.

#	Used clang data			bea-dev (full)
	Orig	MF	ST	F05 orig
0	no	no	no	54.58%
1	yes	no	no	54.61% ± 0.41%
2	yes	no	yes	54.52% ± 0.48%
3	yes	yes	no	54.44% ± 0.58%
4	yes	yes	yes	54.63% ± 0.56%

Table 3: F0.5 on BEA-dev-full for GECToR fine-tuning experiments. For new experiments, average over all seeds ± 2 s.d. is shown.

To evaluate the impact of adding the augmented data to the training dataset, we used an original subset of BEA dev and its augmented manually reviewed versions (described above). The results are shown in table 4 and table 5.

#### 4.4.1 Experiment with singular-they augmentation

We can see that for the baseline model, the gap in F0.5 between the original and augmented (singular-they) version of BEA dev subset is quite significant -9.54%. Fine-tuning on clang8 data led to the shrinking of the gap to -5.86%. We think that this decrease illustrates not an improvement in gender bias, but rather a change in the baseline value due to a shift in precision/recall after fine-tuning. For a more fair comparison, we focused on analyzing the difference between the fine-tuned model on a

#	Used clang data			bea-dev 556		
	Orig	MF	ST	F05 orig	F05 mf_aug	Delta
0	no	no	no	59.23%	58.96%	-0.27%
1	yes	no	no	57.79% ± 0.82%	57.08% ± 1.06%	-0.71%
2	yes	no	yes	57.58% ± 1.12%	57.01% ± 1.2%	-0.57%
3	yes	yes	no	57.88% ± 0.7%	57.33% ± 0.78%	-0.55%
4	yes	yes	yes	58.03% ± 0.76%	57.5% ± 1.04%	-0.53%

Table 4: F0.5 on BEA-dev-556 for GECToR fine-tuning experiments. For new experiments, average over all 10 seeds ± 2 s.d. is shown.

#	Used clang data			bea-dev 195		
	Orig	MF	ST	F05 orig	F05 st_aug	Delta
0	no	no	no	58.28%	48.74%	-9.54%
1	yes	no	no	56.33% ± 2.1%	50.47% ± 1.62%	-5.86%
2	yes	no	yes	55.71% ± 1.22%	54.31% ± 1.62%	-1.4%
3	yes	yes	no	55.77% ± 1.04%	50.33% ± 1.12%	-5.44%
4	yes	yes	yes	56.33% ± 1.48%	54.86% ± 1.46%	-1.47%

Table 5: F0.5 on BEA-dev-195 for GECToR fine-tuning experiments. For new experiments, average over all 10 seeds ± 2 s.d. is shown.

combination of original and augmented data from clang8 (systems 2,3,4) and a model fine-tuned only on original clang data (system 1) (table 5).

We got an improvement in the F0.5 gap for systems 2 and 4 (from -5.86% to -1.4% and -1.47% correspondingly). This reduction is driven by the improvement on the augmented version of bea-dev-195 subset (F0.5 +3.84% and +4.39%) without any (0% for system 4) or with insignificant degradation in quality on the original bea-dev-195 subset (-0.62% for system 3).

We also qualitatively examine the corrections to determine whether explicit instances of bias are reduced through data augmentation (table 6). A linguist manually reviewed model predictions on bea-dev-195-st-aug for systems 1-4 and annotated predictions exhibiting explicit bias, which was defined as pluralization of a referent coreferring with singular *they* or the replacement of singular *they* with a gendered pronoun, or the replacement of *themselves* with *themselves*. System 4 shows the greatest improvement with 7 cases over the baseline of 32. Examples of explicit bias are in Appendix A.

#### 4.4.2 Experiment with feminine/masculine augmentation

For feminine/masculine augmentation, the initial gap between the original subset of BEA (556 sentences) and the augmented version is much smaller -0.71%. Fine-tuning on original and femi-

#	Used clang data			bea-dev-195-st-aug	
	Orig	MF	ST	#	# w/o refl
1	yes	no	no	32	29
2	yes	no	yes	8	7
3	yes	yes	no	34	30
4	yes	yes	yes	7	4

Table 6: Number of sentences displaying explicit bias in bea-dev-195-st-aug. First column is total sentences found to have explicit bias, second is that count minus cases of "themselves">"themselves".

nine/masculine augmentation data (system 3) very slightly reduces this difference only to -0.55%. It's interesting that even singular-they augmentation, without any other gender-swapping, seems to provide a very similar result (difference of -0.57%). However, given the size of confidence intervals, we cannot say that any of our experiments had a significant impact on the gap.

#### 4.4.3 Experiment with both augmentations

Finally, we tried to apply both kinds of augmentation - singular-they and feminine/masculine CDA. The resulting model (system 4) is producing very similar results in terms of gap difference for both BEA subsets that we used: -1.47% on bea-195 (system 2 gap is -1.4%) and -0.53% on bea-556 (system 3 gap is -0.55%), which is showing that

multiple biases might be handled with such a single fine-tuning approach at once. It also seems that augmented training data of two kinds does not interfere with any one evaluation but also does not provide additional benefits from this data interaction.

We believe that there are many other potential possibilities to incorporate augmented data into different stages of the training or change the proportion or the absolute number of original and augmented sentences in training data that might lead to even better improvement with little to no quality degradation. We would like to explore some of them in future work.

## 5 Conclusion

In this work, we developed a novel technique for data augmentation with sentences containing *they* that has an unambiguous singular reference and applied it to the GEC case. We used this technique to help develop a dataset of singular *they* data to parallel data in the BEA shared task dataset that has masculine and feminine pronouns, and with this, we show that GEC systems display bias in their treatment of singular *they* sentences compared to sentences with masculine or feminine pronouns. Additionally, we demonstrated that this technique could be used to reduce bias in GEC systems by fine-tuning the GEC system on the generated synthetic training data.

Because this technique is data-oriented, we believe that it has wider applications, and other NLP systems that display degraded performance with respect to singular *they* may benefit from being trained on data created through this technique.

## Limitations

As noted, this work is limited in that it does not address neopronouns. We speculate that the augmentation techniques deployed in this work may extend to these pronouns as well, we recognize that they do not have the same linguistic reality as *he/she/they* pronouns. Neopronouns may be similar to singular *they* in being relatively infrequent in a naturalistic corpus, but they are also different in that they don't overlap with a frequent morphologically-identical paradigm like plural *they*.

Additionally, the singular *they* augmentation technique we propose is specific to English and distributional facts about English pronouns. For one, English singular *they* morphologically overlaps with a plural pronoun, which is the primary

motivation for using coreference information to identify contexts where *they* would have a primarily singular interpretation. This is often not the case for other languages, as in Swedish where the gender-neutral *hen* is functionally similar to singular *they* but morphologically and distributionally dissimilar in that it does not overlap with a plural pronoun (Gustafsson Sendén et al., 2015).

## Ethics Statement

### Dataset risks

We do not anticipate any risks in releasing the evaluation dataset. This dataset was constructed through the modification of a publicly available dataset commonly used in the evaluation of GEC systems, the dev set of the BEA-2019 shared task (Bryant et al., 2019). These modifications involve the change of gendered words and agreeing verbs to create parallel data across masculine, feminine, and singular *they* pronouns with the goal of evaluating bias in GEC systems. By enabling researchers to measure bias in this way, we believe that the release of this dataset will aid further study in reducing bias in these systems by providing a benchmark.

### Risks of describing data augmentation techniques

We caution that the singular *they* data augmentation technique used in this paper was not designed to generate text that surfaces directly to users. There may be risks to deploying data augmentation techniques at runtime as these techniques are designed to modify gender identity terms; depending on the context of deployment, users may be harmed by such modifications if they result in misgendering or erasure. On the other hand, as we show in this work, use of these techniques to generate training data can reduce bias, and we believe that in this way, the description of this technique will aid in reducing bias in NLP systems.

## Acknowledgements

This research was supported by Grammarly. We thank our colleagues Leonardo Neves, Jade Razzaghi, Yichen Mo, Serhii Yavnyi, and Knar Hovakimyan for their brilliant insights and suggestions over the course of this work. We would also like to thank 3 anonymous reviewers for their helpful comments.

## References

- Lauren Ackerman. 2019. [Syntactic and cognitive issues in investigating gendered coreference](#). *Glossa: a journal of general linguistics*, 4(1). Number: 1 Publisher: Open Library of Humanities.
- Connor Baumler and Rachel Rudinger. 2022. [Recognition of They/Them as Singular Personal Pronouns in Coreference Resolution](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3426–3432, Seattle, United States. Association for Computational Linguistics.
- Bronwyn M. Bjorkman. 2017. [Singular they and the syntactic representation of gender in English](#). *Glossa: a journal of general linguistics*, 2(1). Number: 1 Publisher: Open Library of Humanities.
- Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. [Language \(Technology\) is Power: A Critical Survey of "Bias" in NLP](#).
- Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. [Man is to Computer Programmer as Woman is to Home-maker? Debiasing Word Embeddings](#). Number: arXiv:1607.06520 arXiv:1607.06520 [cs, stat].
- Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. [The BEA-2019 Shared Task on Grammatical Error Correction](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.
- Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. 2022. [Grammatical Error Correction: A Survey of the State of the Art](#). ArXiv:2211.05166 [cs].
- Yang Trista Cao and Hal Daumé III. 2021. [Toward Gender-Inclusive Coreference Resolution: An Analysis of Gender and Bias Throughout the Machine Learning Lifecycle\\*](#). *Computational Linguistics*, 47(3):615–661. Place: Cambridge, MA Publisher: MIT Press.
- Sunipa Dev, Masoud Monajatipoor, Anaelia Ovalle, Arjun Subramonian, J. M. Phillips, and Kai Wei Chang. 2021. [Harms of Gender Exclusivity and Challenges in Non-Binary Representation in Language Technologies](#). In *EMNLP*.
- Hannah Devinney, Jenny Björklund, and Henrik Björklund. 2022. [Theories of "Gender" in NLP Bias Research](#). In *2022 ACM Conference on Fairness, Accountability, and Transparency, FAccT '22*, pages 2083–2102, New York, NY, USA. Association for Computing Machinery.
- Marie Gustafsson Sendén, Emma A. Bäck, and Anna Lindqvist. 2015. [Introducing a gender-neutral pronoun in a natural gender language: the influence of time on attitudes and behavior](#). *Frontiers in Psychology*, 6.
- Chung-hye Han and Keir Moulton. 2022. [Processing bound-variable singular they](#). *Canadian Journal of Linguistics/Revue canadienne de linguistique*, 67(3):267–301. Publisher: Cambridge University Press.
- Lex Konnelly and Elizabeth Cowper. 2020. [Gender diversity and morphosyntax: An account of singular they](#). *Glossa: a journal of general linguistics*, 5(1). Number: 1 Publisher: Open Library of Humanities.
- Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. [The Winograd schema challenge](#). In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning, KR'12*, pages 552–561, Rome, Italy. AAAI Press.
- 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.
- 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:1907.11692 [cs].
- Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. 2019. [Gender Bias in Neural Natural Language Processing](#). arXiv:1807.11714 [cs]. ArXiv: 1807.11714.
- Rowan Hall Maudslay, Hila Gonen, Ryan Cotterell, and Simone Teufel. 2020. [It's All in the Name: Mitigating Gender Bias with Name-Based Counterfactual Data Substitution](#). arXiv:1909.00871 [cs]. ArXiv: 1909.00871.
- Sally McConnell-Ginet. 2013. ["Gender and its relation to sex: The myth of 'natural' gender"](#). In *"Gender and its relation to sex: The myth of 'natural' gender"*, pages 3–38. De Gruyter Mouton.
- Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2011. [Mining Revision Log of Language Learning SNS for Automated Japanese Error Correction of Second Language Learners](#). In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 147–155, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

- Keir Moulton, Chung-hye Han, Trevor Block, Holly Gendron, and Sander Nederveen. 2020. [Singular they in context](#). *Glossa: a journal of general linguistics*, 5(1). Number: 1 Publisher: Open Library of Humanities.
- Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzshanskyi. 2020. [GECToR – Grammatical Error Correction: Tag, Not Rewrite](#). ArXiv:2005.12592 [cs].
- Kostiantyn Omelianchuk, Vipul Raheja, and Oleksandr Skurzshanskyi. 2021. [Text Simplification by Tagging](#). In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 11–25, Online. Association for Computational Linguistics.
- Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. [A Simple Recipe for Multilingual Grammatical Error Correction](#). 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 702–707, Online. Association for Computational Linguistics.
- Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. [Gender Bias in Coreference Resolution](#). *arXiv:1804.09301 [cs]*. ArXiv: 1804.09301.
- Danielle Saunders and Bill Byrne. 2020. [Reducing gender bias in neural machine translation as a domain adaptation problem](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7724–7736, Online. Association for Computational Linguistics.
- Alexey Sorokin. 2022. [Improved grammatical error correction by ranking elementary edits](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11416–11429, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- William Strunk and E. B. White. 1999. *The elements of style*, 4th ed edition. Allyn and Bacon, Boston.
- Tony Sun, Kellie Webster, Apu Shah, William Yang Wang, and Melvin Johnson. 2021. [They, Them, Theirs: Rewriting with Gender-Neutral English](#). ArXiv:2102.06788 [cs].
- Eva Vanmassenhove, Chris Emmery, and Dimitar Shterionov. 2021. [NeuTral Rewriter: A Rule-Based and Neural Approach to Automatic Rewriting into Gender Neutral Alternatives](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8940–8948, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Jun Wang, Benjamin Rubinstein, and Trevor Cohn. 2022. [Measuring and Mitigating Name Biases in Neural Machine Translation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2576–2590, Dublin, Ireland. Association for Computational Linguistics.
- Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. [Gender Bias in Coreference Resolution: Evaluation and Debiasing Methods](#). *arXiv:1804.06876 [cs]*. ArXiv: 1804.06876.

## A Examples of explicit bias

Here we present examples of how the systems investigated here exhibit explicit bias. These are indicated in **bold text**. The list numbers correspond to the systems indicated in Table 3.

Source text:

I said that because when I was eleven my best friend had been operated on **their** back and before the operation he came with me and every day I had to wait for them because they spendt a lot of time in the shower cleaning their long hais , I hated that !

0. I said that because when I was eleven my best friend had been operated on and before the operation he came with me and every day I had to wait for them because they spendt a lot of time in the shower cleaning their long hair . I hated that !
1. I said that because when I was eleven my best friend had been operated on **his** back and before the operation he came with me and every day I had to wait for them because they spendt a lot of time in the shower cleaning their long hair . I hated that !
2. I said that because when I was eleven my best friend had been operated on their back and before the operation he came with me and every day I had to wait for them because they spendt a lot of time in the shower cleaning their long hair . I hated that !
3. I said that because when I was eleven my best friend had been operated on

their back and before the operation he came with me and every day I had to wait for them because they spendt a lot of time in the shower cleaning their long hair . I hated that !

4. I said that because when I was eleven my best friend had been operated on their back and before the operation he came with me and every day I had to wait for them because they spendt a lot of time in the shower cleaning their long hair . I hated that !

Source text:

To take my friend , Taylor , as an example , their homeland were over - developed and become an arid place when they were **a child** .

0. To take my friend , Taylor , as an example , their homeland was over - developed and became an arid place when they were **children** .
1. To take my friend , Taylor , as an example , their homeland was over - developed and became an arid place when they were **children** .
2. To take my friend , Taylor , as an example , their homeland was over - developed and became an arid place when they were a child .
3. To take my friend , Taylor , as an example , their homeland was over - developed and became an arid place when they were **children** .
4. To take my friend , Taylor , as an example , their homeland was over - developed and became an arid place when they were a child .

Source text:

My father has the same program in **their** computer and is able to buy on line tickets w / out leaving the house and moreover w / out picking up the phone .

0. My father has the same program on **his** computer and is able to buy online tickets w / out leaving the house and moreover w / out picking up the phone .

1. My father has the same program on **his** computer and is able to buy online tickets w / out leaving the house and moreover w / out picking up the phone .
2. My father has the same program on their computer and is able to buy online tickets without leaving the house and moreover without picking up the phone .
3. My father has the same program on **his** computer and is able to buy online tickets w / out leaving the house and moreover w / out picking up the phone .
4. My father has the same program on their computer and is able to buy online tickets without leaving the house and moreover without picking up the phone .

## B Error distribution on BEA-dev subsets

### B.1 Edit type distribution on "195" and "556" subsets of bea\_dev

	bea_dev_195				bea_dev_556			
	orig		st_aug		orig		mf_aug	
Edit type	# edits	% edits	# edits	% edits	# edits	% edits	# edits	% edits
R (replacement)	374	66.1%	371	66.0%	814	64.1%	814	64.1%
M (missing)	156	27.6%	157	27.9%	358	28.2%	358	28.2%
U (unnecessary)	36	6.4%	34	6.0%	97	7.6%	96	7.6%
Total	566	100%	562	100%	1269	100%	1268	100%

### B.2 Error categories distribution on "195" and "556" subsets of bea\_dev

	bea_dev_195				bea_dev_556			
	orig		st_aug		orig		mf_aug	
Error category	# edits	% edits	# edits	% edits	# edits	% edits	# edits	% edits
PUNCT	150	26.5%	150	26.7%	314	24.7%	314	24.8%
VERB:TENSE	68	12.0%	66	11.7%	139	11.0%	139	11.0%
OTHER	45	8.0%	48	8.5%	121	9.5%	122	9.6%
PREP	46	8.1%	46	8.2%	115	9.1%	114	9.0%
DET	35	6.2%	34	6.0%	92	7.2%	89	7.0%
ORTH	44	7.8%	44	7.8%	84	6.6%	84	6.6%
SPELL	42	7.4%	41	7.3%	73	5.8%	74	5.8%
VERB	31	5.5%	30	5.3%	67	5.3%	67	5.3%
VERB:FORM	14	2.5%	15	2.7%	36	2.8%	37	2.9%
PRON	10	1.8%	10	1.8%	35	2.8%	36	2.8%
NOUN	17	3.0%	17	3.0%	34	2.7%	35	2.8%
NOUN:NUM	6	1.1%	6	1.1%	33	2.6%	32	2.5%
VERB:SVA	12	2.1%	11	2.0%	25	2.0%	25	2.0%
MORPH	8	1.4%	7	1.2%	24	1.9%	23	1.8%
ADV	12	2.1%	12	2.1%	16	1.3%	17	1.3%
ADJ	10	1.8%	9	1.6%	17	1.3%	16	1.3%
WO	5	0.9%	5	0.9%	12	0.9%	12	0.9%
NOUN:POSS	3	0.5%	3	0.5%	10	0.8%	10	0.8%
PART	5	0.9%	5	0.9%	8	0.6%	8	0.6%
CONTR	1	0.2%	1	0.2%	6	0.5%	6	0.5%
CONJ	2	0.4%	2	0.4%	6	0.5%	6	0.5%
VERB:INFL	0	0.0%	0	0.0%	2	0.2%	2	0.2%
Total	566	100%	562	100%	1269	100%	1268	100%



### C Hyperparameter values for the fine-tuning of GECToR

Hyperparameter name	Hyperparameter value
batch_size	32
accumulation_size	4
n_epoch	5
patience	3
max_len	5
lr	1e-05
cold_steps_count	1
cold_lr	0.001
tp_prob	1
tn_prob	1
updates_per_epoch	10000
special_tokens_fix	1
transformer_model	roberta-base
Pretrained model	
Inference tweaks:	
minimum error probability	0.5
Inference tweaks:	
confidence	0.2

### D Hyperparameter values for the fine-tuning of BART

Hyperparameter name	Hyperparameter value
base_model	BART-Large
src_max_length	80
tgt_max_length	85
beam	2
max_update	16000
loss_criterion	label_smoothed_cross_entropy
optimizer	Adam
weight_decay	0.0
adam_betas	(0.9, 0.98)
adam_eps	1e-06
lr	3e-05

## E Full results of evaluation

System	bea-dev-full			bea-195-orig			bea-195-st-aug			bea-556-orig			bea-556-mf-aug		
	P	R	F05	P	R	F05	P	R	F05	P	R	F05	P	R	F05
GECToR (roberta-base)	64.05%	34.28%	54.57%	70.11%	34.81%	58.28%	56.33%	31.67%	48.74%	69.46%	37.27%	59.23%	68.84%	37.46%	58.96%
BART (seq2seq)	57.46%	39.7%	52.74%	61.32%	42.58%	56.63%	53.59%	39.86%	50.13%	62.73%	46.41%	58.61%	63.05%	46.29%	58.79%
EditScorer (roberta-large)	70.29%	35.77%	58.92%	73.98%	35.16%	60.6%	63.76%	33.81%	54.16%	75%	37.59%	62.55%	73.42%	37.46%	61.59%

Table 7: Scores on BEA-dev subsets for strong GEC baselines.

#	Used clang data			bea-dev (full)		
	Orig	MF	ST	Precision	Recall	F05 orig
0	no	no	no	64.05%	34.28%	54.58%
1	yes	no	no	62.29% $\pm$ 1.3%	36.6% $\pm$ 1.58%	54.61% $\pm$ 0.41%
2	yes	no	yes	62.19% $\pm$ 1.12%	36.35% $\pm$ 0.98%	54.52% $\pm$ 0.48%
3	yes	yes	no	62.38% $\pm$ 1.1%	36.25% $\pm$ 1.34%	54.44% $\pm$ 0.58%
4	yes	yes	yes	62.41% $\pm$ 0.72%	36.46% $\pm$ 0.82%	54.63% $\pm$ 0.56%

Table 8: F0.5 on BEA-dev-full for GECToR fine-tuning experiments. For new experiments, average over all seeds  $\pm$  2 s.d. is shown.

# ReadAlong Studio Web Interface for Digital Interactive Storytelling

Aidan Pine<sup>1</sup> David Huggins-Daines<sup>2</sup> Eric Joanis<sup>1</sup> Patrick Littell<sup>1</sup> Marc Tessier<sup>1</sup>  
Delasie Torkornoo<sup>3</sup> Rebecca Knowles<sup>1</sup> Roland Kuhn<sup>1</sup> Delaney Lothian<sup>1</sup>

<sup>1</sup>National Research Council Canada, Ottawa ON, Canada  
first.last@nrc-cnrc.gc.ca

<sup>2</sup>Independent Researcher dhd@ecolingui.ca

<sup>3</sup>Algonquian Dictionaries and Language Resources Project, Carleton University  
Ottawa ON, Canada delasie.torkornoo@carleton.ca

## Abstract

We develop an interactive web-based user interface for performing text–speech alignment and creating digital interactive “read-along” audio books that highlight words as they are spoken and allow users to replay individual words when clicked. We build on an existing Python library for zero-shot multilingual text–speech alignment (Littell et al., 2022), extend it by exposing its functionality through a RESTful API, and rewrite the underlying speech recognition engine to run in the browser. The ReadAlong Studio Web App is open-source, user-friendly, prioritizes privacy and data sovereignty, allows for a variety of standard export formats, and is designed to work for the majority of the world’s languages.

## 1 Introduction

A “read-along”, as seen in Figure 1, is an interactive language tool that highlights words as they are spoken and allows users to replay certain words when clicked (Luchian and Junker, 2004). Language learners are able to interact with these multimodal text/audio documents by repeating the pronunciation of specific words, pausing at a specific place in the document, and following along visually as the text is spoken. This tool promotes reading and listening skills in language learners, which are target skills that are underrepresented in language-learning technology (Shadiev and Yang, 2020).

While the ReadAlong Studio Web App is compatible with many languages (§2.1), it was designed specifically to support learners in an Indigenous language revitalization context, where listening comprehension is often a key priority (Hermes et al., 2012; Lothian et al., 2019). In addition to being more beneficial for comprehension than reading or listening alone (Webb and Chang, 2022), reading while listening can also help promote listening-based skills such as auditory discrimination (i.e., the ability to discriminate between sounds) (Chang,



Figure 1: A screenshot of a web component read-along published for Atikamekw. Other read-alongs published for Atikamekw can be found at <https://atikamekw.atlasling.ca/lecture-audio/>. Highlighting guides the reader to the word currently being spoken in the recording, and the reader can play single words by clicking on them.

2009). Furthermore, read-alongs could be used to promote speaking skills by using them in conjunction with speaking tasks, such as shadowing (i.e., reading/speaking along with fluent speech while trying to match pace) (Kadota, 2019).

Building read-alongs, however, can be challenging. Aligning text and speech manually requires a considerable amount of time, and requires some expertise in using audio software. On the other hand, while text–speech alignment can be automated (e.g., Schiel, 1999; Gorman et al., 2011; McAuliffe et al., 2017; Kürzinger et al., 2020), these systems require non-trivial expertise in speech technology and machine learning to train and deploy a model for a new language (MacKenzie and Turton, 2020).

The zero-shot text–speech aligner described in Littell et al. (2022) partially addresses this issue; it can align speech and text in a new language without having seen any prior data in that language. However, it is still a command-line tool that, for full use of its capabilities, requires some familiarity with text and XML formats; it is still not something the average language teacher could use without significant training.

Thus, we decided to develop a web-based graphical user interface on top of that system. In the process, we ported a significant amount of the system to JavaScript so that audio processing could happen entirely in the browser. This allows users to create their own read-alongs without requiring them to write code or install anything on their computers, and without sending any audio data to a third party. The software is free and open source, and has a data privacy policy designed to affirm community data sovereignty.

## 1.1 ReadAlong Studio Web App

The ReadAlong Studio Web App is designed as a two-step process for creating the kind of read-alongs seen in Figure 1. First, the user either writes or uploads some text, records or “uploads” audio (the audio is not actually uploaded to a server, but kept in memory in the browser; see §2.5), and selects the language of their data. The actual text–speech alignment is performed automatically, and the user is taken to step two. Step two presents the read-along to the user in a WYSIWYG<sup>1</sup>-inspired editable mode and lets them add a title and a subtitle, images for each page, and translations for each line, as desired.<sup>2</sup>

There are public resources that instruct users on how to work with the ReadAlong Studio Web App. One such resource is embedded within the ReadAlong Studio Web App as a “tour” that guides users through the steps of creating a read-along (see Figure 3 in Appendix A). We encourage the interested reader to explore the interface themselves,<sup>3</sup> review some of the screenshots available in Appendix A, or the publicly available documentation associated with our recent workshop session at the 8<sup>th</sup> International Conference on Language Documentation & Conservation in March 2023.<sup>4</sup>

## 2 Design Decisions

### 2.1 Language Agnostic

It is well-known that language technologies are not equally available to the world’s languages. Reviews of studies on language-learning technologies

have found that not only are target languages typically restricted to European and majority languages (Shadiev and Yang, 2020; Burston, 2014), but over half of technologies researched target English, with that percentage increasing within the last decade (Sauro, 2016). While there are active efforts to promote more linguistic diversity among language technologies, sociohistorical and socioeconomic factors are still the most significant determinants of whether the language you speak is supported by the technologies you use. Additionally, for the NRC’s Indigenous Languages Technology project, on which the majority of the authors work, our mandate is to support many languages (Kuhn et al., 2020). Thus, our goal was to make a web-interface for creating read-alongs that would be accessible in many languages with as few modifications as possible.

As mentioned in §1, other high-quality text–audio alignment tools exist, but the Littell et al. (2022) aligner best suited our needs since it not only supports zero-shot alignment in 39 (mostly Canadian Indigenous) languages out of the box, but also supports zero-shot alignment of most languages through the use of a rough, language-neutral “fallback” G2P engine (see Littell et al., 2022; Pine et al., 2022, for further details). This tends to work well on languages with relatively transparent orthographies that use characters in cross-linguistically common ways, but will potentially run into trouble in languages that use characters in uncommon ways or have significant orthographic ambiguities; we discuss this in greater detail in the Limitations section. However, in a series of workshops (§3) and in follow-up communication with users, it appears that all the languages users have tried so far have been successful, even those with unique orthographies like Korean and Western Armenian.

Choosing a zero-shot aligner had profound effects on the ReadAlong Studio Web App, since the interface only needs to handle the inference step. Unlike the Elpis tool (Foley et al., 2018) designed for the more challenging task of general speech recognition and transcription, there is no training of a model on user data, and we thus avoid the complication of guiding the user through this process.

The ReadAlong Studio Web App interface itself is also language agnostic in the sense that it has been written using Angular’s built-in translation/internationalization library, with the site cur-

<sup>1</sup>What You See Is What You Get

<sup>2</sup>This “editing” mode is also available outside of the ReadAlong Studio Web App in any read-along by changing the “mode” attribute on the custom read-along HTML element from “VIEW” to “EDIT” (see Figure 2 in Appendix A).

<sup>3</sup><https://readalong-studio.mothers-tongues.org/>

<sup>4</sup><https://readalongs.github.io/ICLDC-Docs/>,  
<https://github.com/ReadAlongs>

rently available in English, French, and Spanish; the code can be adapted to other languages, and further contributions are welcome.

## 2.2 Portability

While read-alongs can be visualized within the ReadAlong Studio Web App, the purpose of them is to be shared and deployed in a variety of places. To make the interactive read-along user interface as transferable as possible between different web frameworks, we implemented it with StencilJS,<sup>5</sup> a framework for building custom elements using the Web Component open standard API.<sup>6</sup> Further information on how to embed a read-along in any website can be found in Figure 2 in Appendix A.

StencilJS is able to build wrappers around the web components, allowing for greater interoperability with modern web frameworks like Angular, React or Vue.<sup>7</sup> We currently build and publish an Angular wrapper on npm<sup>8</sup> and will publish React and Vue integrations if there is a demand.

ReadAlong Studio also generates a self-contained HTML file that Base64-encodes and embeds all the multimedia content into a single file that can be viewed in any browser, even when Internet access is unavailable. This is an important consideration for rural communities without ubiquitous WiFi and mobile data, where teachers send multimedia content to students via SD cards or USB drives, or where students download content to devices at a central location like a school.

The ReadAlong Studio Web App, for creating read-alongs, is also fairly portable: the software is open source with a permissive license (MIT), its dependencies are all open source, and it can be deployed with minimal resource requirements, albeit with some IT expertise. The front end is a static web page written using Angular, that can be served locally, or at no cost on a service like GitHub Pages. The back end is an API written in Python with FastAPI<sup>9</sup> that can be run locally, or on any cloud server with as little as 512 MB of RAM.

The various options for local, internal network, and cloud deployment, as well as the public deployment we provide, enable communities to choose the solution that best meets their accessibility and

privacy requirements.

## 2.3 Implementation of zero-shot alignment

As in Littell et al. (2022), alignment is done by performing highly constrained finite-state grammar recognition using an English acoustic model and a dictionary generated by roughly mapping the output of zero-shot G2P to the target phoneset. The acoustic model is the same one used in Pocket-Sphinx (Huggins-Daines et al., 2006), and is thus a very old technology optimized for efficiency over accuracy.

The recognizer itself<sup>10</sup> is compiled into WebAssembly using Emscripten<sup>11</sup> and wrapped in a hand-coded JavaScript API. By avoiding the use of C++ in the wrapper and removing functionality irrelevant to the web environment, we obtain a code footprint of 214KB of WebAssembly and 40KB of (minimized) JavaScript. The model is downloaded asynchronously after loading the page, and recognition is done asynchronously in the main browser thread, entirely on the user’s computer.

Compared to the original system in Littell et al. (2022), we use a much smaller acoustic model (to limit the download size to 10MB) and also down-sample the audio to 8kHz to speed up processing. This typically results in a decrease of 1-3 points in the F1 score of alignments, but we feel that the improved responsiveness and reduced network traffic, along with the privacy and sovereignty considerations detailed in §2.5, make up for what is generally an imperceptible difference.

In future work we plan to further rewrite the aligner to use more modern acoustic modeling and decoding technology, if this can be done while also maintaining or reducing the storage and memory footprint.

## 2.4 Targeting Open Formats

From the user’s perspective, choosing to work with a particular technology comes with risks, including whether you will be compromising your intellectual property or rights to privacy by using the tool (see §2.5) and whether the time you spend using the tool or creating resources within it will be “locked-in” to the platform. We have heard many stories from teachers and curriculum developers of times they have invested hundreds of hours of work creating content in particular sites/products only to find

<sup>5</sup><https://stenciljs.com/docs/introduction>

<sup>6</sup><https://www.webcomponents.org/introduction>

<sup>7</sup><https://angular.io/>, <https://react.dev/>, <https://vuejs.org/>

<sup>8</sup><https://www.npmjs.com/package/@readalongs/ngx-web-component>

<sup>9</sup><https://fastapi.tiangolo.com/>

<sup>10</sup><https://github.com/ReadAlongs/SoundSwallower>

<sup>11</sup><https://emscripten.org>

that the company goes bankrupt or switches to a different monetization strategy, rendering their content lost, inaccessible, or locked into proprietary file formats.

We do not intend for the ReadAlong Studio project to end unexpectedly, or for any of the core contributors to suddenly become unavailable; however, these outcomes are rarely anticipated for any project. To prevent a situation where the technology becomes unmaintained and users are unable to use the software, we have implemented a variety of features to ensure users' creations can persist beyond the life of this particular project.

The choice of programming languages, format and other software dependencies were carefully considered to give this application a longer than average “shelf life”. Firstly, the software is released through a permissive open-source license which will hopefully encourage a diverse community of developers to take part in maintaining the software—with efforts being shared across all users. We expect the default HTML output format to continue to be usable on any JavaScript-enabled HTML5-compatible browser. The raw text and audio *could* be extracted from this, but it would take some technological expertise. To make storage and archiving more accessible and prevent “vendor lock-in”, every stage of the pipeline offers downloads to standard formats. Text written directly into the software can be downloaded as .TXT, audio recorded can be downloaded as .MP3, and the resulting alignments can be downloaded as Praat TextGrids,<sup>12</sup> ELAN files,<sup>13</sup> and WebVTT<sup>14</sup> or SRT<sup>15</sup> subtitles; formats which have wide-spread support across many software tools.

## 2.5 Privacy & Data Sovereignty

Deciding to use a particular technology often comes with consequences for the user's privacy and ownership over their data (Keegan, 2019). Globally, there is a history of theft and misuse of Indigenous language data by academic researchers and external collaborators. In response, Indigenous communities in Canada have created language authorities and data sovereignty principles, such as the First Nations Principles of ownership, control, access,

and possession (OCAP®)<sup>16</sup> (First Nations Information Governance Centre, 2023). It is against principles like these for Indigenous language data to be owned or kept in any part by external organizations. Since the motivation for the ReadAlong Studio Web App is primarily to support language education within a language revitalization context, we wanted to develop a tool that affirmed community efforts to remain in control of their data.

In order to adhere to these principles, we prioritized having alignments created locally on the user's machine. Part of our ability to do this stems from the fact that we chose a zero-shot text-audio alignment method, which requires no training data; the only data it requires is the data to be turned into the read-along. As described in §2.3, the second author of this paper also re-implemented the speech recognition engine in WebAssembly and JavaScript so that all alignments happen in the user's browser.

There is still one part of the process that does not occur locally, however: the G2P engine required for the zero-shot method to work is written in Python, so the text for the read-along is uploaded to a remote server for G2P processing. However, the text is not stored on the server after processing, and, as discussed in §2.2, a community could deploy the backend on their own servers if there is a need for greater privacy. Our eventual goal is to implement a version of the G2P engine that will also run in the browser.

We also prioritized user privacy with respect to collecting user analytics. In order to obtain a better understanding of how users are using the site, we have implemented analytics using Plausible Analytics ([plausible.io](https://plausible.io)): a privacy-focused analytics solution that does not use cookies or track individuals, but rather presents aggregate data about user operating systems, viewport size, and custom-specified “actions”. These actions tell us what percentage of users actually create a read-along once they visit the site, or which output file format they download (§2.4). This information is included in our privacy policy on the ReadAlong Studio Web App; we allow users to opt out from the analytics at their discretion.

## 3 Discussion & Usage

We held two 90 minute workshop sessions titled “Watch me Speak! Interactive Storytelling using

<sup>12</sup><https://www.fon.hum.uva.nl/praat/>

<sup>13</sup><https://archive.mpi.nl/tla/elan>

<sup>14</sup><https://www.w3.org/TR/webvtt1/>

<sup>15</sup><https://en.wikipedia.org/wiki/SubRip>

<sup>16</sup>OCAP® is a registered trademark of the First Nations Information Governance Centre (FNIGC)

ReadAlong Studio” at the 8<sup>th</sup> International Conference on Language Documentation & Conservation to walk potential users through the ReadAlong Studio Web App in March 2023. The final twenty minutes of each workshop were dedicated to a “language party”, inspired by the Aikuma Project’s initiative of the same name:<sup>17</sup> participants were invited to create a read-along for their language and share it with the group. Participants created and shared read-alongs in Crow (Siouan), Halkomelem and Nsyilxcən (Salishan), Michif, Gitksan (Tsimshianic), Quechua, Korean, Nuuchah-Nulth (Wakashan), Paiwan (Austronesian), Sáliba (Piaroa–Saliban), Takelma, and Western Armenian (Indo-European).

Following the workshops, a participant shared that they had been nervous to create and share their read-along during the language party session of the workshop because they were worried they might not spell things correctly (since they did not have the language-specific keyboard installed), and potentially cause the system to break or not function properly. They were pleasantly surprised when the words of their text became highlighted with their voice. While we built ReadAlong Studio Web App to be *language* agnostic, it is also agnostic to dialect and orthographic variations, which are very common for a variety of reasons in Indigenous language revitalization contexts in Canada (see §5 of Littell et al., 2017). Many tools that are created for a language revitalization context do not offer such generalized support for different dialects or writing systems, potentially systematically excluding certain users. By contrast, the ReadAlong Studio Web App’s tolerance for variations in pronunciation and spelling shows potential for fostering a non-judgemental environment for learners to practise speaking their language.

According to Plausible Analytics, of the 525 unique visitors from 23 countries to visit the ReadAlong Studio Web App in the first two months after the launch on February 26, 2023, 122 users created 396 read-alongs, with 65 users going on the tour and 56 users downloading their read-alongs. Among the 56 unique users that downloaded read-alongs, they downloaded read-alongs 174 times in the various available formats. The most popular format (which is also the default) was the offline HTML version. While the total number of people creating read-alongs is still modest, we are encour-

aged by the amount those users are interacting with the tool and we believe that these statistics demonstrate achievement of our goal for users to be able to create read-alongs for many languages without requiring technological expertise.

## 4 Conclusion

In this paper, we detailed the motivation for, and design decisions of, web-based software for creating read-alongs titled ReadAlong Studio Web App. As key design considerations, we highlighted support for many languages (§2.1), portability and longevity (§2.2), avoiding vendor lock-in (§2.4), and affirming privacy and data sovereignty concerns (§2.5). Future work involves improving the workflow for correcting and adjusting alignments, increasing language support, and refactoring the G2P engine to JavaScript for complete client-side processing.

## Limitations

**Accessibility** We have tried to develop ReadAlong Studio Web App with accessibility in mind, using accessible colour contrasts, ensuring buttons have aria-labels, and ensuring that the website is legible when zoomed-in to 200%, among other considerations. Using Google PageSpeed Insights, our website scores 89 for Accessibility, but we recognize that there are still improvements to be made; specifically, we would like to perform an audit of the website with respect to Web Content Accessibility Guidelines (WCAG).

**Inexact transcription** ReadAlong Studio will work best if the transcription is exact; that is, if there are as few discrepancies between the text and audio as possible. If extraneous text exists (such as page numbers, chapter titles, or translations), or if the audio includes un-transcribed speech (such as false starts), these errors will accumulate and can result in poor alignments.

The extent to which these discrepancies affect the final result depends on the length of the recording to be aligned. In practice, we have found that ReadAlong Studio is able to recover from minor transcription errors when the speech data to be aligned are around 5 minutes or less in length. We have successfully aligned much longer (up to 40 minute) files, but “your mileage may vary” depending on the exactness of the transcription, the language’s orthography, and the type of data used.

<sup>17</sup><https://www.languageparty.org/>

**Singing** Several teachers have successfully aligned songs with the corresponding text using ReadAlong Studio. For such an alignment to be successful, however, it is necessary that the sung words be vocalized clearly, and not be drowned out by the accompanying music (if any). Extended legato singing (e.g., where one syllable is extended across multiple notes) can also cause poor alignments, since the speech-trained acoustic model does not expect single syllables to correspond to multiple intensity peaks in this way.

**Language support** The software works with most languages out-of-the-box. As mentioned in §2.1, ReadAlong Studio comes with support for 39 languages built-in, and handles other languages with a rough, best-guess G2P based on Unicode table information. At several international workshops (§3), we found that it worked reasonably well with every language brought by workshop participants, even those with unique alphabets like Western Armenian or Korean.

However, not every language will work equally well. It will typically work well in languages with systematic orthographies that use letters in cross-linguistically common ways. We anticipate difficulty with orthographies that use familiar letters in cross-linguistically unusual ways, such as “font-encodings” (Pine and Turin, 2018), abjads that leave out many vowels, and languages like Japanese where the pronunciation of logographs is highly variable and determined by context. Just like a human could not simply guess the missing vowels in written Hebrew without knowing Hebrew, the software will not be able to do this either.

Additionally, the software is limited to languages which are both written and spoken—we do not support signed languages since the aligner requires audio to align with text, and the tool is fundamentally inapplicable to unwritten languages.

The interface itself is currently only translated in English, French, and Spanish, limiting potential users who do not speak one of those languages.

**Numbers and symbols** While ReadAlong Studio can do rough zero-shot G2P for most alphabetic and syllabic writing systems, it is not capable of general text normalization—while it can guess that “T” might be pronounced [t] in an unfamiliar language, it simply has no basis to guess any particular pronunciation for “634”, as this task is not only language-dependent but highly variable within any

given language (Bigi, 2011). Therefore, all input must be “spelled out” for alignment to succeed.

If the input contains numbers or symbols, ReadAlong Studio Web App will prompt the user with a warning that it found uninterpretable symbols.<sup>18</sup>

## Ethics Statement

We have addressed a wide variety of ethical concerns throughout the paper, including trying to ensure that the tool supports a diverse audience, does not create technological dependency, and affirms First Nations research principles of ownership, control, access, and possession (OCAP®).

In the preceding **Limitations** section, we have also tried to be transparent in the ways that our tool might not be adequate for certain users. A final outstanding ethical concern of ours is that our software could potentially be misused to create and distribute content that does not belong to the content creator, causing a variety of potential harms. In the workshop series that we ran, we highlighted this by explicitly warning participants against making read-alongs without first ensuring they had obtained the proper permissions and consent to build and/or distribute content created with audio or text that does not belong to them. While we have not designed a system that is able to ensure this type of misuse will not happen, we are trying to mitigate this risk by explicitly warning against this type of misuse in our public messaging related to the software.

We have attempted to be thorough in considering ethical issues related to our software, but we are aware that our considerations are not comprehensive. We encourage prospective users of the ReadAlong Studio Web App, and indeed of any language technology, to be mindful of this, and to think critically about the possible risks and benefits that come with using any particular tool. We direct interested readers to the excellent “Check Before You Tech”<sup>19</sup> checklist, and welcome any further related questions from current or prospective users.

## Acknowledgements

This work would not have been possible without the many collaborators who shared their ex-

<sup>18</sup>This does impact languages like Skwxwú7mesh sníchim, in which “7” is a valid orthographic character representing a glottal stop. In this case, “7” could be converted to a glottal stop if the mapping existed, otherwise it would be skipped.

<sup>19</sup><https://fpcc.ca/resource/check-before-you-tech/>



pertise, precious recordings, and experience using the ReadAlong Studio Web App Interface, including but not limited to the Yukon Native Language Centre, the Kitigan Zibi Cultural Centre, WSÁNEĆ School Board, the Pirurvik Centre, Conseil de la Nation Atikamekw, Onkwawenna Kentyohkwa, Owennatekha Brian Maracle, Silver Rae Stevens, Timothy Montler, Marie-Odile Junker, Hilaria Cruz, Nathan Thanyehtenhas Brinklow, Francis Tyers, Fineen Davis, Eddie Antonio Santos, Mica Arseneau, Vasilisa Andriyanets, Christopher Cox, Bradley Ellert, Robbie Jimerson, Shankhalika Srikanth, Sabrina Yu, Jorge Rosés Labrada, Caroline Running Wolf, Michael Running Wolf, Fangyuan (Toby) Huang, Zachery Hindley, Darrel Schreiner, Luyi Xiao, Siqi Chen, Kwok Keung Chung, Koon Kit Kong, He Yang, Yuzhe Shen, Rui Wang, Zirui Wang, Xuehan Yi, and Zhenjie Zhou.

## References

- Brigitte Bigi. 2011. [A multilingual text normalization approach](#). In *2nd Less-Resourced Languages workshop, 5th Language & Technology Conference*.
- Jack Burston. 2014. The reality of MALL: Still on the fringes. *CALICO Journal*, 31(1):103–125.
- Anna C-S Chang. 2009. Gains to L2 listeners from reading while listening vs. listening only in comprehending short stories. *System*, 37(4):652–663.
- First Nations Information Governance Centre. 2023. [The First Nations principles of OCAP®](#).
- Ben Foley, Joshua T Arnold, Rolando Coto-Solano, Gautier Durantin, T Mark Ellison, Daan van Esch, Scott Heath, Frantisek Kratochvil, Zara Maxwell-Smith, David Nash, et al. 2018. Building speech recognition systems for language documentation: The CoEDL endangered language pipeline and inference system (ELPIS). In *SLTU*, pages 205–209.
- Kyle Gorman, Jonathan Howell, and Michael Wagner. 2011. Prosodylab-aligner: A tool for forced alignment of laboratory speech. *Canadian Acoustics*, 39(3):192–193.
- Mary Hermes, Megan Bang, and Ananda Marin. 2012. [Designing Indigenous language revitalization](#). *Harvard Educational Review*, 82(3):381–402.
- David Huggins-Daines, Mohit Kumar, Arthur Chan, Alan W Black, Mosur Ravishankar, and Alexander I Rudnicky. 2006. PocketSphinx: A free, real-time continuous speech recognition system for hand-held devices. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 1, pages I–I. IEEE.
- Shuhei Kadota. 2019. *Shadowing as a practice in second language acquisition*. Routledge, New York, NY.
- Te Taka Keegan. 2019. [Issues with Māori sovereignty over Māori language data](#).
- Roland Kuhn, Fineen Davis, Alain Désilets, Eric Joanis, Anna Kazantseva, Rebecca Knowles, Patrick Littell, Delaney Lothian, Aidan Pine, Caroline Running Wolf, Eddie Santos, Darlene Stewart, Gilles Boulianne, Vishwa Gupta, Brian Maracle Owenatékha, Akwiratékha’ Martin, Christopher Cox, Marie-Odile Junker, Olivia Sammons, Delasie Torkornoo, Nathan Thanyehténhas Brinklow, Sara Child, Benoît Farley, David Huggins-Daines, Daisy Rosenblum, and Heather Souter. 2020. [The Indigenous Languages Technology project at NRC Canada: An empowerment-oriented approach to developing language software](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5866–5878, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Ludwig Kürzinger, Dominik Winkelbauer, Lujun Li, Tobias Watzel, and Gerhard Rigoll. 2020. CTC-segmentation of large corpora for German end-to-end speech recognition. In *Speech and Computer: 22nd International Conference, SPECOM 2020, St. Petersburg, Russia, October 7–9, 2020, Proceedings 22*, pages 267–278. Springer.
- Patrick Littell, Eric Joanis, Aidan Pine, Marc Tessier, David Huggins Daines, and Delasie Torkornoo. 2022. [ReadAlong Studio: Practical zero-shot text-speech alignment for Indigenous language audio-books](#). In *Proceedings of the 1st Annual Meeting of the ELRA/ISCA Special Interest Group on Under-Resourced Languages*, pages 23–32, Marseille, France. European Language Resources Association.
- Patrick Littell, Aidan Pine, and Henry Davis. 2017. [Waldayu and Waldayu Mobile: Modern digital dictionary interfaces for endangered languages](#). In *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 141–150, Honolulu. Association for Computational Linguistics.
- Delaney Lothian, Gokce Akcayir, and Carrie Demmans Epp. 2019. Accommodating Indigenous people when using technology to learn their ancestral language. *Proceedings of the first International Workshop on Supporting Lifelong Learning co-located with the 20th International Conference on Artificial Intelligence (AIED 2019)*, pages 16–22.
- Radu Luchian and Marie-Odile Junker. 2004. [Developing an on-line Cree read-along with syllabics](#). *Carleton University Cognitive Science Technical Report*.
- Laurel MacKenzie and Danielle Turton. 2020. [Assessing the accuracy of existing forced alignment software on varieties of British English](#). *Linguistics Vanguard*, 6(s1):20180061.

- Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. 2017. Montreal Forced Aligner: Trainable text-speech alignment using Kaldi. In *Interspeech*, volume 2017, pages 498–502.
- Aidan Pine and Mark Turin. 2018. Seeing the Heiltsuk orthography from font encoding through to Unicode: A case study using convertextract. In *Proceedings of the LREC 2018 Workshop CCURL 2018*, pages 27–30.
- Aidan Pine, Patrick William Littell, Eric Joanis, David Huggins-Daines, Christopher Cox, Fineen Davis, Eddie Antonio Santos, Shankhalika Srikanth, Delasie Torkornoo, and Sabrina Yu. 2022. [G<sub>i</sub>2P<sub>i</sub>: Rule-based, index-preserving grapheme-to-phoneme transformations](#). In *Proceedings of the Fifth Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 52–60, Dublin, Ireland. Association for Computational Linguistics.
- Shannon Sauro. 2016. Does CALL have an English problem? *Language Learning & Technology*, 20(3):1–8.
- F. Schiel. 1999. Automatic phonetic transcription of nonprompted speech. In *Proc. of the ICPHS*, pages 607–610.
- Rustam Shadiev and Mengke Yang. 2020. [Review of studies on technology-enhanced language learning and teaching](#). *Sustainability*, 12(2).
- Stuart Webb and Anna C-S Chang. 2022. How does mode of input affect the incidental learning of collocations? *Studies in Second Language Acquisition*, 44(1):35–56.

## Appendix A: Screenshots

Figure 2 illustrates how simple it is to insert a read-along into a web page. Figure 3 shows the guided tour users can follow to better understand how to use the Studio. Figure 4 expands the drop-down menu for choosing download formats. Figures 5 and 6 are screen captures of the two-step ReadAlong Studio Web App interface.

```

<!DOCTYPE html>
<html>
  <head></head>
  <body>
    <!-- Here is how you declare the Web Component. These files are produced by ReadAlong Studio, the
         paths must point where they are hosted. Multiple ReadAlongs can exist on the same page. -->
    <read-along href='my-file.readalong' audio='my-file.mp3' mode='VIEW' />
  </body>

  <!-- Import the package at the end of your HTML file. The example here is using the unpkg CDN. -->
  <script src='https://unpkg.com/@readalongs/web-component'></script>
</html>

```

Figure 2: Minimal HTML code required to embed a read-along in your website: insert the read-along element where the read-along should be displayed, and add the script link at the end of the HTML source to load the code required for rendering the read-along.

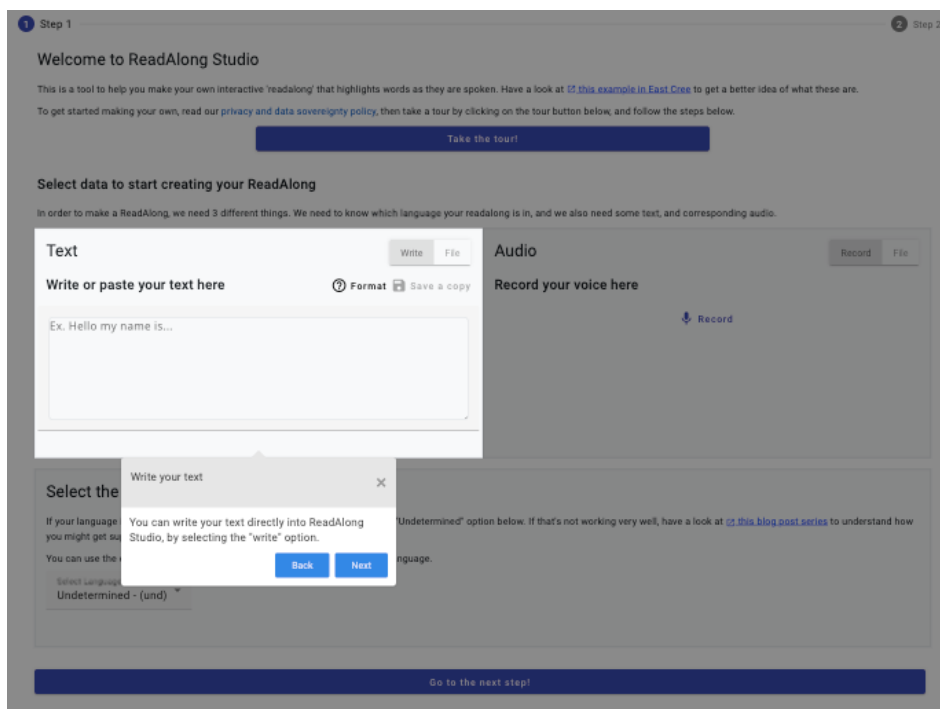


Figure 3: Guided tour in ReadAlong Studio demonstrating how to write text for creating a read-along

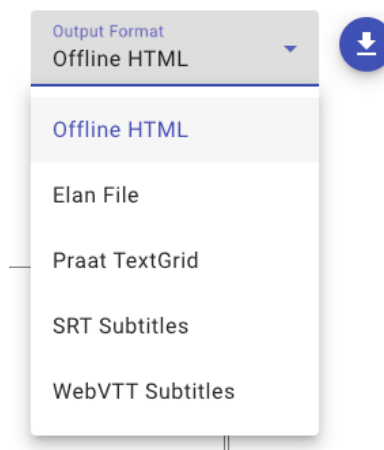


Figure 4: Drop-down menu showing the variety of downloadable output formats in ReadAlong Studio.

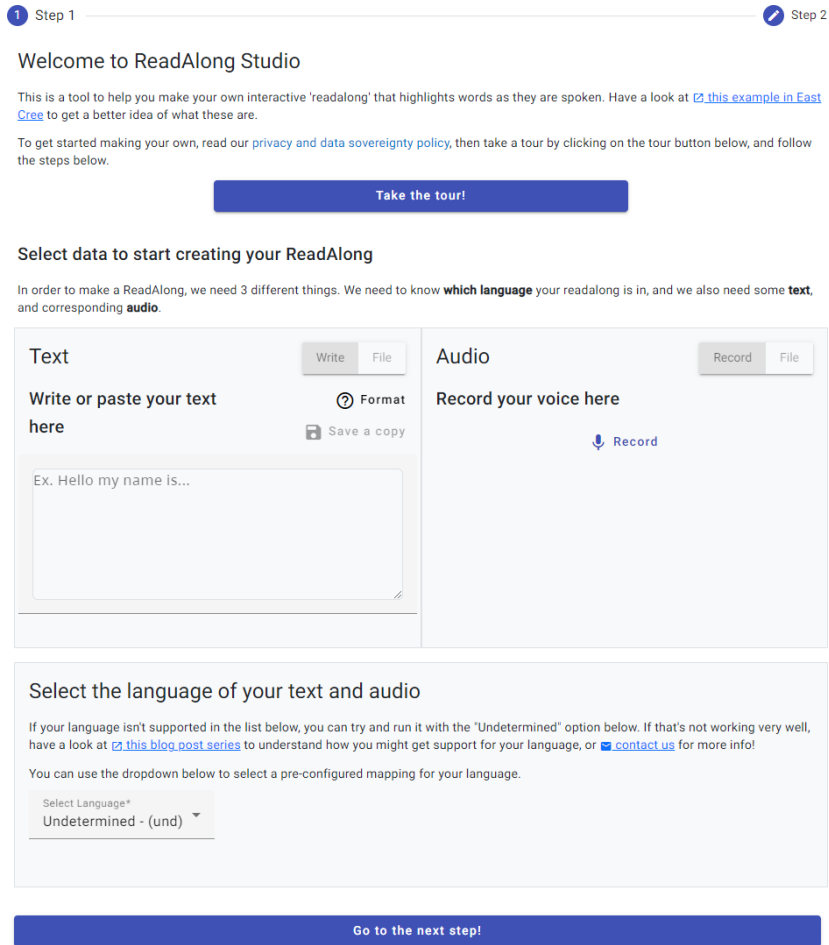


Figure 5: Step 1 of the two-step ReadAlong Studio Web App interface: selecting text and audio



Figure 6: Step 2 of the two-step ReadAlong Studio Web App interface: editing the created read-along

# Labels are not necessary: Assessing peer-review helpfulness using domain adaptation based on self-training

**Chengyuan Liu**

North Carolina State University  
cliu32@ncsu.edu

**Divyang Doshi**

North Carolina State University  
ddoshi2@ncsu.edu

**Muskaan Bhargava**

North Carolina State University  
mbharg5@ncsu.edu

**Ruixuan Shang**

University of North Carolina at Chapel Hill  
rshang@unc.edu

**Jialin Cui**

North Carolina State University  
jcui9@ncsu.edu

**Dongkuan Xu**

North Carolina State University  
dxu27@ncsu.edu

**Edward Gehring**

North Carolina State University  
efg@ncsu.edu

## Abstract

A peer-assessment system allows students to provide feedback on each other’s work. An effective peer assessment system urgently requires helpful reviews to facilitate students to make improvements and progress. Automated evaluation of review helpfulness, with the help of deep learning models and natural language processing techniques, gains much interest in the field of peer assessment. However, collecting labeled data with the “helpfulness” tag to build these prediction models remains challenging. A straightforward solution would be using a supervised learning algorithm to train a prediction model on a similar domain and apply it to our peer review domain for inference. But naïvely doing so can degrade the model performance in the presence of the distributional gap between domains. Such a distributional gap can be effectively addressed by Domain Adaptation (DA). Self-training has recently been shown as a powerful branch of DA to address the distributional gap. The first goal of this study is to evaluate the performance of self-training-based DA in predicting the helpfulness of peer reviews as well as the ability to overcome the distributional gap. Our second goal is to propose an advanced self-training framework to overcome the weakness of the existing self-training by tailoring knowledge distillation and noise injection, to further improve the model performance and better address the distributional gap.

## 1 Introduction

Peer review is a learning tool that enables students to evaluate their peers’ assignments or projects (Gamage et al., 2021; Topping, 2009; Li et al., 2019). It can help instructors enhance their teaching (Çevik et al., 2015; Gamage et al., 2021), and allow students to develop skills in assessing and providing feedback to others. Figure 1 illustrates the steps of the peer review process. It starts with the authors submitting their work. The peers then evaluate the work and provide both textual feedback and numerical scores. The author assesses the feedback and tends to accept only the helpful reviews to make further revisions (Lundstrom and Baker, 2009). The instructors can refer the numerical scores provided by the reviewers to give the final grades. Therefore, identifying helpful peer reviews can enhance the benefits to students from the peer-review process (Nelson and Schunn, 2009; Ramachandran et al., 2017). Automatic recognition of peer-review helpfulness has been studied limitedly with the help of deep learning models and natural language processing (Xiong and Litman, 2011b; Xiao et al., 2022). However, in order to create a reliable model that can accurately predict helpfulness, a considerable amount of peer-review data labeled with helpfulness is required (Chapelle et al., 2009). The students receiving the reviews are the most suitable individuals to label the data, but the difficulty in collecting labeled reviews from stu-

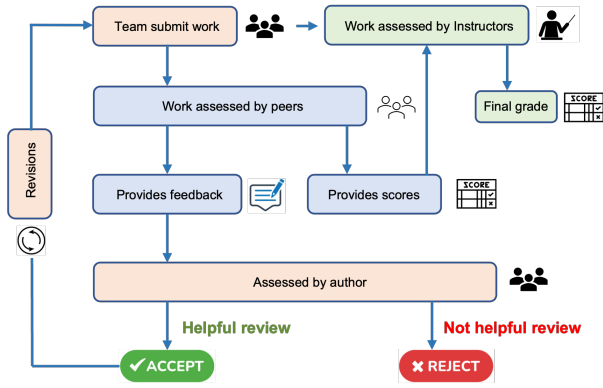


Figure 1: Peer review process flowchart. The pipeline involves the evaluation of the feedback from peers by the author. Only the helpful review is accepted and taken into account for further revisions.

dents poses a challenge. Moreover, the subjective nature of “helpfulness” creates ambiguity, making it challenging to achieve a consensus in a team on whether a review is helpful. As a result, obtaining sufficient labeled data to develop a robust model for predicting helpfulness remains a significant obstacle.

A straightforward solution to overcome the challenges of collecting labeled data is to adapt a model trained on a pre-existing labeled dataset from a similar domain that includes “helpfulness” tags to our peer review domain. Specifically, we can train the prediction model on a “**source domain**” labeled data following the supervised manner, and generate “helpfulness” prediction on the “**target domain**” unlabeled data from our peer reviews. However, the discrepancies in data distribution between the source and target domains, i.e., **domain shift** (Liu et al., 2020; Wang and Breckon, 2020), can cause the model’s performance to degrade on unseen target domain data.

In this paper, in order to address the domain shift issue, we propose to apply **Self-training** (a.k.a., Pseudo-labeling) (Zou et al., 2019; Lee, 2013; Feng et al., 2021; Mei et al., 2020; Yu et al., 2021), as a promising technique in Domain Adaptation (DA) ((Ben-David and Uner, 2012; Liu et al., 2021; Zou et al., 2019)). Self-training-based Domain Adaptation aims to transfer knowledge learned from the source domain to the target domain, by involving the unlabeled data from the target domain in the model training. We hypothesize that learning from the unlabeled data can enhance the generalization ability, and facilitate the effective knowledge transfer across domains. This hypothesis will be validated through the experiment results.

Our proposed approach for domain adaptation using self-training follows the “student-teacher model” framework (Pu and Li, 2023). As shown in Figure 3, the student and teacher models will constantly exchange their roles during the iterative process, and the student model will continuously learn from the pseudo labels predicted by the teacher model. Self-training helps to overcome the domain shift between the source and target domains (Liu et al., 2021). As a novelty, our study also proposes an advanced self-training framework that utilizes knowledge distillation (Hinton et al., 2015) and noise injection (Xie et al., 2020) techniques to overcome some weaknesses of the traditional self-training, and further improve the adaptation performance. By incorporating knowledge distillation, the student model can better mimic the teacher model and break through the limitation of only being able to learn from the “hard labels” provided by the teacher model. Additionally, the incorporation of noise injection enables the student model to outperform the teacher model by learning from the augmented data, which is beyond what the teacher model predicts.

The contributions are summarized as follows:

- We propose the use of self-training-based domain adaptation to predict peer review helpfulness, which overcomes the challenge of collecting labeled data and mitigates the domain shift issue.
- We improve self-training by tailoring knowledge distillation techniques and utilizing soft labels to provide more comprehensive knowledge for the student model to learn from the teacher model.
- We improve self-training by introducing noise during the student model training phase, enabling the student model to learn beyond the predictions generated by the teacher model.

## 2 Related Work

### 2.1 Peer Review Helpfulness Prediction

Previous peer-review research has not paid much attention to helpfulness prediction, with only a few studies utilizing NLP techniques to identify key features in review comments to evaluate the quality. Xiong and Litman (2011a) conducts a pioneering study on predicting peer-review helpfulness and suggests that techniques used in other domains can

be applied to the peer-review domain. Zingle et al. (2019) describes a method for automatically detecting *suggestions* in review text. Xiao et al. focus on detecting *problem statements* which point out the problems that need to be addressed in review comments.

However, there is no study that directly investigates predicting helpfulness based on the semantics of the review content. The lack of labeled training data also poses a challenge to building such a prediction model, due to the subjective nature of helpfulness and controversies surrounding its definition. Xiong and Litman (2011b) reports that there is a great deal of variation among students and even domain experts in terms of “what constitutes a helpful comment.”

Fortunately, several researchers (Tsur and Rapoport, 2009; Qu et al., 2020; Yang et al., 2015) have explored predicting the helpfulness of online product reviews, which can be conveniently labeled with “helpfulness” through user voting from online shopping platforms. In this study, we adapt the task of predicting the helpfulness of online product reviews to our academic peer reviews, drastically reducing the need for collecting peer-review labeled data.

## 2.2 Domain Adpatation

Training models on the “source domain”(with labeled data) and testing them on the “target domain”(without labeled data) using supervised learning algorithms often fail due to the distributional gap between the two domains, commonly known as domain shift (Long et al., 2015).

Domain adaptation (DA) aims to alleviate the effect of domain shift. Various methods have been proposed to mitigate that by aligning the source and target domain in the feature space. These approaches explicitly align their statistics or use adversarial learning. For instance, Glorot et al. (2011) proposed an autoencoder-based domain adaptation network, which extracts high-level representations from both source and target domain data. They then trained a linear classifier to learn from the source data’s extracted features and applied it to the target data. Long et al. (2015) used a deep neural network to learn transferable features across domains by adding multiple adaptation layers to the task-specific representations. They match the marginal distributions of both domains. Furthermore, Ganin and Lempitsky (2015) proposed an

adversarial-based domain adaptation approach that adds an effective Gradient Reversal Layer (GRL) to the model, inspired by *Generative Adversarial Networks* (Goodfellow et al., 2014), to match the domain gap.

Despite the success of the existing approaches, Ben-David and Uner (2012) highlighted the difficulty of applying the above feature-adaptation-based approaches in DA and suggested that none of those methods have the capacity to generalize well to the unlabeled target domain data. In this study, we propose to use self-training (a.k.a. *pseudo-labeling*) as a promising alternative to the feature-adaptation approaches to better handle the domain shift.

## 2.3 Self-training

Self-training is a popular technique in semi-supervised learning, where a supervised method is applied for classification or regression tasks in a semi-supervised manner. In self-training, the model is trained on a small amount of labeled data, then it generates predictions on the unlabeled data, which are adopted as pseudo-labels. The model is retrained on the combination of both labeled data and pseudo-labeled data, and the process iterates until convergence.

In pioneering work, Lee (2013) first introduces the classical pseudo-labeling method, which differs from the self-training framework in that the model is not retrained after each pseudo-labeling. He et al. (2020) successfully applies the self-training framework in NLP tasks such as machine translation and text summarization, also provides a comprehensive evaluation of its effectiveness. Another approach proposed by Pu and Li (2023) is the self-training framework with a “student-teacher model”, in which a teacher model assigns pseudo-labels to unlabeled data, and a student model is trained on the combined dataset iteratively. However, the vanilla self-training suffers from certain limitations of the student model’s learning abilities, which we defined as “inability to learn sufficiently from the teacher model” and “inability to learn beyond the teacher model”.

To address these limitations, we propose applying knowledge distillation and noise injection to the self-training framework, which ensures a well-performing student model. Our approach improves the student model’s learning ability, achieving decent results over the traditional self-training ap-

proach.

### 3 Methodology

#### 3.1 Self-training for Domain Adaptation

Self-training for domain adaptation is a bit different from the traditional single-domain self-training approach, the workflow is illustrated in Figure 3 and formulated using the following steps:

**Requirements:** Source-domain labeled dataset  $D_{SL} = \left\{ \left( x_i^L, y_i \right) \right\}_{i=1}^{N_{sl}}$  and target-domain unlabeled dataset  $D_{TU} = \left\{ \left( x_j^U \right) \right\}_{j=1}^{N_{tu}}$  where  $N_{sl}$  and  $N_{tu}$  stands for the number of samples in source and target dataset respectively;  $x_i^L$  and  $x_j^U$  are the vector representations of each review text; and  $y_i$  stands for the one-hot encoding label for source domain labeled data.

**Steps:**

1. To initiate the self-training process, a teacher model  $f_\tau(\theta_*)$  (e.g., a BERT-based language classification model (Devlin et al., 2019)) is trained on the labeled dataset from the source domain, to minimize the cross-entropy loss using Equation 1.

$$\frac{1}{N_{sl}} \sum_{i=1}^{N_{sl}} CE(y_i, f_\tau(x_i^L, \theta)) \quad (1)$$

2. The teacher model is then used to generate pseudo-labels on the unlabeled dataset from the target domain, as shown in Equation 2.

$$\hat{y}_j = f_\tau(x_j^U, \theta_*), \forall j \in [1, N_{tu}] \quad (2)$$

3. A student model  $f_s(\theta')$  (e.g., BERT-based language classification model) is then learned to minimize the cross entropy loss on a combined dataset  $D_C = \{(x_c)\}_{c=1}^{N_c}$ , which includes the source domain labeled data  $D_{SL}$  and target domain pseudo-labeled data  $D_{TU}$ . The loss is calculated using Equation 3.

$$\frac{1}{N_c} \sum_{c=1}^{N_c} CE(y_c, f_s(x_c, \theta')) \quad (3)$$

where  $N_c = N_{sl} + N_{tu}$ ,  $(x_c, y_c)$  represents  $(x_i, y_i)$  and  $(x_j, \hat{y}_j)$  for the source labeled set and the target pseudo-labeled set, respectively.

#### 3.2 Knowledge Distillation — “Student Learns More From Teacher”

Knowledge Distillation (KD) is a technique for compressing a model by using a more complex teacher model that has already been trained to guide a smaller, less-complex student model. This is done to maintain the accuracy of the original teacher model while reducing the model size and computational resources required (Hinton et al., 2015).

In traditional classification, the model aims to map input features to the one-hot labels, which only provide class information. However, with KD, the teacher model can generate a continuous distribution of class labels (i.e., soft labels) for each sample, allowing for more information to be used. The student model is then trained to closely match the output distribution of the teacher model.

Specifically, KD employs softmax probability to generate soft labels. In contrast, traditional classification tasks use cross-entropy as the loss function, with hard one-hot labels as targets. However, as highlighted by Hinton et al. (2015), this approach can result in the loss of valuable information on the similarity between and within classes. By using the probability output from the softmax layer instead, KD is able to retain more information.

Incorporating the KD technique into our self-training framework aims to improve the performance of the student model by acquiring additional knowledge from the pseudo-labels generated by the teacher model. Figure 3 illustrates the process of knowledge distillation in self-training. In this process, we retained both the hard and soft pseudo-labels generated by the teacher model to preserve an adequate amount of information. Consequently, we substituted the conventional cross-entropy loss function with the KD loss function (Hinton et al., 2015) as represented in Equation 4.

$$L = - \sum_{i=1}^K p_i^{hard} \log q_i + \sum_{i=1}^K p_i^{soft} \log \left( \frac{p_i^{soft}}{q_i} \right) \quad (4)$$

The first segment of the equation calculates the cross-entropy loss between the hard pseudo-labels  $p_i^{hard}$  (one-hot encoding), which are generated by the teacher model and represented through one-hot encoding, and the soft output  $q_i$  produced by the student model. The latter part computes the Kullback–Leibler divergence (Wikipedia contributors, 2023) between the soft pseudo-labels  $p_i^{soft}$  from the teacher model and the output  $q_i$  of the student model. Our objective is to account for both the



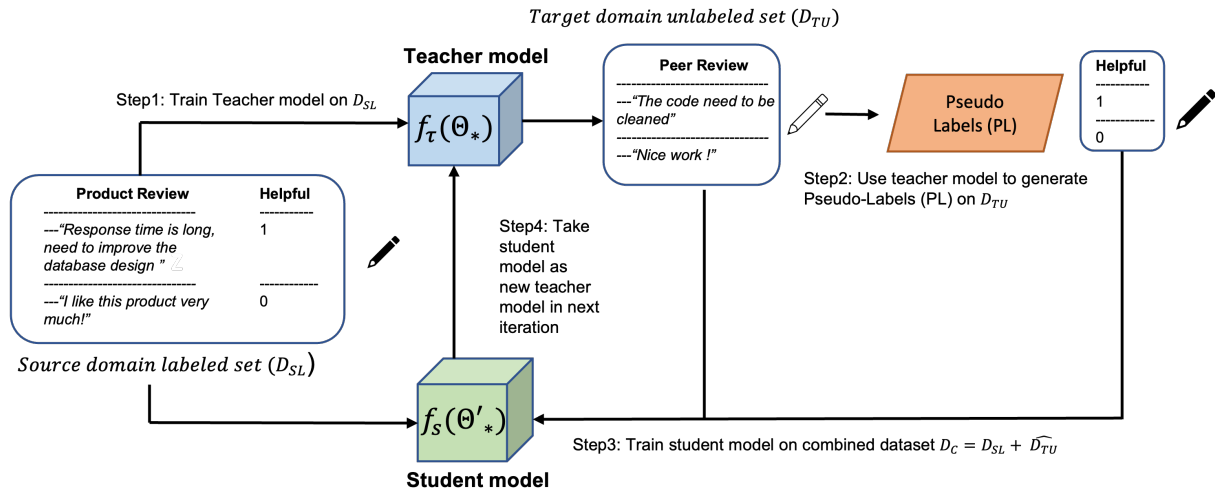


Figure 2: Self-training pipeline for peer review helpfulness detection across domains. A “Teacher model” will be trained on the labeled data from the source domain. Then a “Student model” will be trained using both the labeled data from the source domain and the pseudo-labeled data from the target domain labeled by the teacher model. The trained “Student model” will be used as the new “Teacher model” in the next iteration.

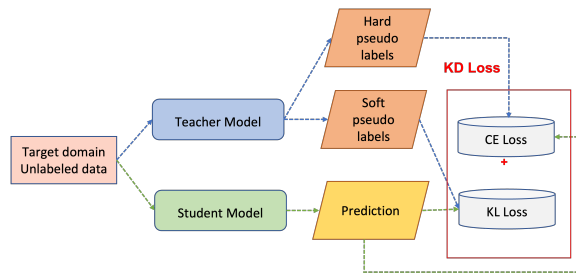


Figure 3: Schematic diagram of the KD loss computation in single self-training iteration

hard and soft pseudo-labels’ information while calculating the loss.

### 3.3 Noise Injection – “Student Learns Beyond Teacher”

The use of Knowledge Distillation enables the student model to learn more information from the soft labels. However, it is crucial to acknowledge that the primary objective of employing KD is to train a smaller and more efficient student model that has the same capabilities as the teacher model. Conversely, in self-training, our goal is to train a superior performing student model. To achieve this, we must ensure that the student model is not less complex than the teacher model and has the ability to capture more variance of the data. Unfortunately, incorporating KD is insufficient to accomplish this.

Noise injection creates a more challenging environment for the student model to learn beyond the predictions. In this study, we utilize data augmentation as the noise injection method in the stu-

dent model training phase. We implement back-translation (Ng et al., 2019) as a prominent text-augmentation approach on the target domain’s pseudo-labeled data. For the augmented data, we keep the same pseudo-labels (both hard and soft). Consequently, this requires the student model to ensure that a translated version of the text yields the same output as the original text, which is also known as *consistency regularization* (Ho et al., 2022). By doing this, we improve the student model by providing augmented data to learn beyond what the teacher model predicts.

## 4 Experiments and Results

### 4.1 Datasets

**Source Domain Labeled Data.** Our source domain labeled data is obtained from the *Amazon Product Review* (Ni et al., 2019), which contains 29 categories of online products. Since the categories’ relevance to our peer-review data varies, we conduct experiments on two product categories. The “software” category is chosen, since it is closely related to our peer-review data, as both involve user-experience feedback on developed applications. The “automotive” category is also selected to evaluate whether data from a less-relevant domain would impact the performance of domain adaptation. Additionally, we create two datasets of varying size within each category and investigate how significantly the size affects the performance.

Our objective is to predict binary class labels of

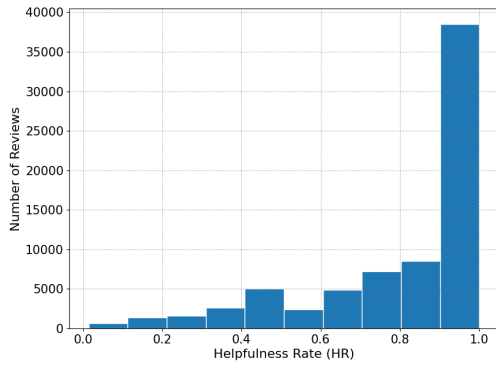


Figure 4: Helpfulness rate distribution of “software” product review. Note in these plots that the majority of the reviews have the “helpfulness ratio” larger than 0.8.

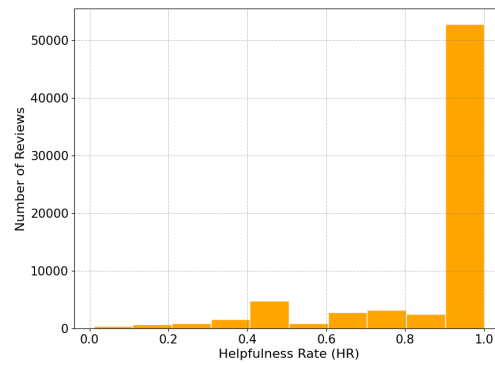


Figure 5: Helpfulness rate distribution of “automotive” product review.

reviews, where “0” represents “not helpful” and “1” represents “helpful”. However, the original data contain the “helpfulness” tags, which have been collected through user votes formatted as: “the number of users who find the review helpful out of the total number of users who vote for the review” (e.g., [2,3] implies that out of 3 users who voted on the review, 2 of them rated it as helpful, thus the “helpfulness ratio (hr)” is  $2/3$ ). To convert this into binary class labels, we decide to set a threshold for the “helpfulness ratio” and split the data into the two classes of “helpful” and “not helpful”. Figures 4 and 5 illustrate the distribution of the “helpfulness ratio” for “software” and “automotive” datasets. To create a clear distinction between the two classes, we choose the reviews with a “helpfulness ratio” above 0.85 as helpful and below 0.35 as unhelpful reviews.

After text cleaning and processing, we collect 500 and 2000 labeled product reviews for each of the two categories. We also ensured that the class labels are evenly distributed.

**Target Domain Unlabeled Data** The peer review data of the target domain is collected from the Expertiza system (Gehring et al., 2006), which is a web-based peer review system used in a masters-level computer science class. The system requires students to review assignments from their peers and provide numerical scores and textual feedback. We extract the textual feedback data from the fall semesters of 2017 to 2020, resulting in 24,619 review samples after cleaning and processing.

**Target Domain Validation Data.** We should also need a validation set from the target domain to assess whether it generalized well by using our

proposed self-training approach. However, collecting “helpfulness” tags in our peer review system is challenging. Fortunately, the Expertiza system (Gehring et al., 2006) provides a way for students to tag the reviews they received as having or not having particular characteristics. These tags identified features such as *contains problem statement* and *contains suggestion*. A study conducted by Xiao et al. (2022) states that these two features are highly correlated with review helpfulness. Therefore, we decide to utilize these tags as a proxy for “helpfulness” tag to create our target domain validation data.

We generated the “helpfulness” label for our validation sets by considering review comments tagged as containing *both* “problem statement” and “suggestion” as “helpful” and those without either of these two as “not helpful”. (Comments containing either “problem statement” or “suggestion” tags, but not both, were excluded from the dataset.) The result was a balanced validation set of 7000 reviews, consisting of an equal number of “helpful” and “not helpful” samples.

## 4.2 Experiment Settings

**Supervised Learning Baseline** The first baseline method uses a supervised learning approach. We aim to investigate the existence of a domain shift in our task. We applied the pre-trained “bert-base-uncased” model from the Hugging Face library (Wolf et al., 2019) and fine-tuned it on the labeled dataset from the source domain. Then, we validated its performance on the target domain validation set. The domain shift is evaluated by calculating the accuracy score of the model on the validation set. The detailed settings of this baseline

Parameters	Value
Tokenizer	'bert-based-uncased'
Classification model	'bert-based-uncased'
Number of classes	2
Loss function	Cross-entropy loss
Optimizer	Adam
Dropout	0.3
Learning rate	2e-5
Epoch	5
Batch size	16

Table 1: Supervised learning baseline experiment setting

are presented in Table 1.

**Self-training Baseline** We establish our second baseline as applying the vanilla self-training approach to examine whether learning from the target domain unlabeled data could enhance the performance and address the domain shift. As shown in Figure 3, the training of the teacher model uses the exact same settings as the supervised learning baseline presented in Table 1. Afterward, the self-training loop is initiated, where each loop starts by generating pseudo-labels using the trained teacher model and ends by taking the trained student model as the new teacher model. In the self-training phase, we have set the value of *outer\_epoch* to 10, which indicates how many times we will repeat the loop described above. Additionally, we also set the value of *inner\_epoch* to 3, which represents the number of training iterations of the student model in each self-training loop.

**Our Proposed Approach** To overcome the limitations of the self-training, we propose an approach that integrates *knowledge distillation* and *noise injection* in the self-training loop. The core idea behind knowledge distillation is to generate soft pseudo-labels in the form of prediction probabilities to enable student models to learn from additional knowledge. Therefore, in addition to retaining the prediction probabilities from the teacher model, we also replace the cross-entropy loss with the “kd\_loss” (defined in Equation 4) for training the student model. However, we continue to use the cross-entropy loss for training the teacher model with hard labels. Consequently, the general loss function of both the student and teacher models can be formulated as follows:

$$loss = \alpha \times KL\_loss + (1 - \alpha) \times CE\_loss \quad (5)$$

in which we introduce an  $\alpha$  value to regulate the weight of the KL divergence loss and the cross-entropy loss. We set  $\alpha$  to 0 to exclusively use the cross-entropy part in the teacher model training. In

contrast, during student model training, we set  $\alpha$  to 0.5 to consider both parts of the loss with soft and hard pseudo-labels. It would be interesting as future work to experimentally search for an optimal value of  $\alpha$  to explore its impact on performance.

To add the noise injection part, we utilize the pre-trained EN-DE/DE-EN and EN-RU/RU-EN back-translation models (Ng et al., 2019). Considering that transformer-based augmentation models can exponentially increase the computation time, we limit the amount of data to be augmented at 40% by setting the augmentation ratio to 0.4.

### 4.3 Experiment Results

The experimental results are presented in Table 2, where we evaluate the performance of our proposed approach, by measuring the accuracy on the validation dataset and comparing it with the baseline approaches. To analyze the results, we aim to answer the following research questions:

**RQ1: Does domain shift exist in our task?**

According to the first row of Table 2, training the model on product reviews and using it to predict peer reviews leads to very poor results. The accuracy scores are mostly around 50%, and some are even worse than random guessing. This suggests that the domain shift does exist in our case, and without applying any domain adaptation techniques, the model’s performance will be poor.

**RQ2: Is the performance different for different categories of product reviews?**

In addition to assessing the existence of domain shift in our task, we are also interested in investigating the extent to which domain shift differed across various categories of product review data. Table 2 shows that the category “software” product review, which is more relevant to the peer review domain, yields better results than the “automotive” review. For example, when using the same 2000 labeled data, training on the “software” category yields 55.1% accuracy with the supervised learning baseline, while only 43.83% accuracy is achieved on the “automotive” category. After applying our proposed approach, we achieve 68.52% accuracy on “software” over 48.80% on “automotive” data. Hence, we conclude that source domain data with different relevance to the peer review data will result in varying degrees of the distributional gap, which is a crucial factor in domain-adaptation tasks.

**RQ3: Does self-training mitigate domain shift by leveraging unlabeled data from the target**

	Amazon “Software” data		Amazon “Automotive” data	
	500 labeled data	2000 labeled data	500 labeled data	2000 labeled data
<i>Supervised Learning</i>	60.34%	55.1%	41.02%	43.83%
<i>Self-training</i>	60.67%	66.64%	42.31%	43.3%
<b><i>Our Approach</i></b>	<b>63.05%</b>	<b>68.52%</b>	<b>52.54%</b>	<b>48.61%</b>

Table 2: Accuracy scores of the proposed approach on various source domain labeled datasets

### domain?

Examining the second row of Table 2, fairly good improvements can be observed by applying self-training. In addition to an average improvement of 3.16% in accuracy across all datasets, the greatest improvement of 11.54% is achieved with 2000 labeled “software” reviews. This convincingly demonstrates the benefits of learning from unlabeled target-domain data, even in the absence of labeled information. The results indicate considerable effectiveness of using self-training to tackle domain shift issues.

### RQ4: Is our proposed approach able to enhance the performance of self-training?

We aim to assess whether our proposed approach is able to improve performance and overcome the limitations of the self-training baseline. The third row of Table 2 shows that our approach, which incorporates knowledge distillation and noise injection, outperforms the self-training baseline. We achieved the best accuracy score of 68.52%, the greatest improvement of 10.42%, and an average improvement of 4.95% over the self-training baseline. These results demonstrate that by incorporating knowledge distillation and noise injection, the student model learns more effectively and outperforms the teacher model.

### RQ5: Does the effectiveness of the proposed approach depend on the size of the source-domain labeled dataset?

We perform experiments using different sizes of labeled datasets from the source domain. As presented in Table 2, the “software” dataset shows better performance with 2000 labeled reviews compared to 500 labeled reviews. Surprisingly, we find that for the “automotive” reviews, training with only 500 labeled reviews outperforms even 2000 labeled reviews. We hypothesize that with a less relevant source domain dataset, a larger labeled dataset can result in more misleading training due to a larger distributional gap. Furthermore, our pro-

	“Software” labeled data	“Automotive” labeled data
Self-training + kd	67.14%	44.87%
Self-training + noise	<b>68.9%</b>	43.03%
<b>Our proposed approach</b>	68.52%	<b>48.61%</b>

Table 3: Comparison of the accuracy scores by applying KD and noise injection respectively with self-training.

posed approach shows a greater improvement over the self-training baseline with 500 labeled reviews than with 2000 labeled reviews of both categories. This indicates that our approach is more effective in improving self-training, given that only limited data can be gleaned from the source domain.

### 4.4 Ablation Study

In addition to the results presented in Table 2, we also examine the effect of each individual component in our proposed approach on the overall performance. We conduct extensive experiments by using only knowledge distillation or noise injection. The results are evaluated with the 2000 labeled reviews from both categories, which are shown in Table 3.

The table reveals some intriguing findings. We unexpectedly achieve a better result than our proposed approach by using only the noise injection, trained on the “software” labeled data. This indicates that using both components together may cause a performance drop. Similarly, we observe that using KD alone leads to better performance compared to noise injection alone, for the “automotive” review dataset. This contrasts with our finding for the “software” data. In the future, we plan to explore ways to optimize the use of both components and make them mutually beneficial.

## 5 Conclusion

This study first highlights the pedagogical significance of predicting helpful reviews in peer assessment to benefit student learning, and then considers the challenge of collecting labeled data to build a reliable prediction model. We explore a solution via domain adaptation to reduce the need of collecting labeled data. Our primary contribution is proposing self-training as an optimal domain-adaptation technique to address the domain-shift issue that commonly arises when transferring knowledge between domains. Furthermore, we incorporate knowledge distillation and noise injection into self-training to improve performance. The experimental results exhibit promise in utilizing self-training and show the effectiveness of our proposed approach. In addition, we discuss future work in optimizing the integration of knowledge distillation and noise injection.

## References

- Shai Ben-David and Ruth Urner. 2012. [On the hardness of domain adaptation and the utility of unlabeled target samples](#). volume 7568 LNAI, pages 139–153.
- Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. 2009. Semi-supervised learning. *IEEE Transactions on Neural Networks*, 20:542.
- 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.
- Hao Feng, Minghao Chen, Jinming Hu, Dong Shen, Haifeng Liu, and Deng Cai. 2021. [Complementary pseudo labels for unsupervised domain adaptation on person re-identification](#). In *IEEE Transactions on Image Processing*, volume 30, pages 2898–2907.
- Dilrukshi Gamage, Thomas Staubitz, and Mark Whiting. 2021. [Peer assessment in moocs: Systematic literature review](#). *Distance Education*, 42:268–289.
- Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15*, page 1180–1189. JMLR.org.
- Edward Gehringer, Luke Ehresman, Susan G GConger, and Prasad Wagle. 2006. [Reusable learning objects through peer review: The expertiza approach](#). volume 3, pages 1–2.
- Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on International Conference on Machine Learning, ICML'11*, page 513–520, Madison, WI, USA. Omnipress.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. [Generative adversarial nets](#). In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.
- Junxian He, Jiatao Gu, Jiajun Shen, and Marc’ Aurelio Ranzato. 2020. [Revisiting self-training for neural sequence generation](#). In *International Conference on Learning Representations*.
- Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. [Distilling the knowledge in a neural network](#).
- Stella Ho, Ming Liu, Lan Du, Yunfeng Li, Longxiang Gao, and Shang Gao. 2022. [Semi-supervised continual learning with meta self-training](#). In *Proceedings*

- of the 31st ACM International Conference on Information and Knowledge Management, CIKM '22, page 4024–4028, New York, NY, USA. Association for Computing Machinery.
- Dong-Hyun Lee. 2013. [Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks](#). pages 1–6.
- Bo Li, Yezhen Wang, Tong Che, Shanghang Zhang, Sicheng Zhao, Pengfei Xu, Wei Zhou, Yoshua Bengio, and Kurt Keutzer. 2020. [Rethinking distributional matching based domain adaptation](#). *arXiv preprint arXiv:2006.13352*.
- Hongli Li, Yao Xiong, Charles Vincent Hunter, Xiuyan Guo, and Rurik Tywoniw. 2019. [Does peer assessment promote student learning? a meta-analysis](#). *Assessment and Evaluation in Higher Education*, 45:1–19.
- Hong Liu, Jianmin Wang, and Mingsheng Long. 2021. [Cycle self-training for domain adaptation](#).
- Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. 2015. Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15*, page 97–105. JMLR.org.
- Kristi Lundstrom and Wendy Baker. 2009. [To give is better than to receive: The benefits of peer review to the reviewer's own writing](#). *Journal of Second Language Writing*, 18:30–43.
- Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. 2020. Instance adaptive self-training for unsupervised domain adaptation. In *European Conference on Computer Vision (ECCV)*.
- Melissa M. Nelson and Christian D. Schunn. 2009. [The nature of feedback: How different types of peer feedback affect writing performance](#). *Instructional Science*, 37:375–401.
- Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission. In *Proc. of WMT*.
- Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. [Justifying recommendations using distantly-labeled reviews and fine-grained aspects](#). 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 188–197, Hong Kong, China. Association for Computational Linguistics.
- Xiaokun Pu and Chunguang Li. 2023. [Meta-self-training based on teacher-student network for industrial label-noise fault diagnosis](#). *IEEE Transactions on Instrumentation and Measurement*, 72:1–11.
- Xianshan Qu, Xiaopeng Li, Csilla Farkas, and John Rose. 2020. An attention model of customer expectation to improve review helpfulness prediction. In *Advances in Information Retrieval*, pages 836–851, Cham. Springer International Publishing.
- Lakshmi Ramachandran, Edward F. Gehringer, and Ravi K. Yadav. 2017. [Automated assessment of the quality of peer reviews using natural language processing techniques](#). *International Journal of Artificial Intelligence in Education*, 27:534–581.
- Keith J. Topping. 2009. [Peer assessment](#). *Theory into Practice*, 48:20–27.
- Oren Tsur and Ari Rappoport. 2009. [Revrank: A fully unsupervised algorithm for selecting the most helpful book reviews](#). *Proceedings of the International AAAI Conference on Web and Social Media*, 3(1):154–161.
- Qian Wang and Toby Breckon. 2020. Unsupervised domain adaptation via structured prediction based selective pseudo-labeling. volume 34, pages 6243–6250.
- Wikipedia contributors. 2023. Kullback–leibler divergence — Wikipedia, the free encyclopedia.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.
- Yunkai Xiao, Tianle Wang, Xinhua Sun, Yicong Li, Yang Song, Jialin Cui, Qinjin Jia, Chengyuan Liu, and Edward F. Gehringer. 2022. [Modeling review helpfulness with augmented transformer neural networks](#). In *2022 IEEE 16th International Conference on Semantic Computing (ICSC)*, pages 83–90.
- Yunkai Xiao, Gabriel Zingle, Qinjin Jia, Shoaib Akbar, Yang Song, Muyao Dong, Li Qi, and Edward Gehringer. Problem detection in peer assessments between subjects by effective transfer learning and active learning. In *Proceedings of the 13th International Conference on Educational Data Mining (EDM 2020)*.
- Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. [Unsupervised data augmentation for consistency training](#). volume 33, pages 6256–6268. Curran Associates, Inc.
- Wenting Xiong and Diane Litman. 2011a. [Automatically predicting peer-review helpfulness](#). pages 502–507. Association for Computational Linguistics.
- Wenting Xiong and Diane Litman. 2011b. Understanding differences in perceived peer-review helpfulness using natural language processing. In *Proceedings of the Sixth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 10–19.

- Yinfei Yang, Yaowei Yan, Minghui Qiu, and Forrest Bao. 2015. Semantic analysis and helpfulness prediction of text for online product reviews. 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 38–44.
- Fei Yu, Mo Zhang, Hexin Dong, Sheng Hu, Bin Dong, and Li Zhang. 2021. [Dast: Unsupervised domain adaptation in semantic segmentation based on discriminator attention and self-training](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10754–10762.
- Gabriel Zingle, Balaji Radhakrishnan, Yunkai Xiao, Edward Gehringer, Zhongcan Xiao, Ferry Pramudianto, Gauraang Khurana, and Ayush Arnav. 2019. Detecting suggestions in peer assessments. In *Proceedings of the 12th International Conference on Educational Data Mining (EDM 2019)*, pages 474–479. International Educational Data Mining Society.
- Yang Zou, Zhiding Yu, Xiaofeng Liu, B. V.K.Vijaya Kumar, and Jinsong Wang. 2019. [Confidence regularized self-training](#). volume 2019-Octob, pages 5981–5990.
- Yasemin Demiraslan Çevik, Tülin Haşlaman, and Serkan Çelik. 2015. [The effect of peer assessment on problem solving skills of prospective teachers supported by online learning activities](#). *Studies in Educational Evaluation*, 44:23–35.

# Generating Dialog Responses with Specified Grammatical Items for Second Language Learning

Yuki Okano<sup>1</sup>, Kotaro Funakoshi<sup>1</sup>, Ryo Nagata<sup>2,3</sup>, Manabu Okumura<sup>1,3</sup>

<sup>1</sup>Tokyo Institute of Technology

<sup>2</sup>Konan University

<sup>3</sup>RIKEN Center for Advanced Intelligence Project

{okano, funakoshi, oku}@lr.pi.titech.ac.jp

nagata-bea2023@ml.hyogo-u.ac.jp.

## Abstract

This paper proposes a new second language learning task of generating a response including specified grammatical items. We consider two approaches: 1) fine-tuning a pre-trained language model (DialoGPT) by reinforcement learning and 2) providing a few-shot prompt to a large language model (GPT-3). For reinforcement learning, we examine combinations of three reward functions that consider grammatical items, diversity, and fluency. Our experiments confirm that both approaches can generate responses including the specified grammatical items and that it is crucial to consider fluency rather than diversity as the reward function.

## 1 Introduction

The use of dialog systems for language learning has attracted attention. Many studies have introduced dialog systems as training partners for language learners and verified their effectiveness. According to previous studies (Kim, 2016; Tegos et al., 2014; Ruan et al., 2019), the advantages of using dialog systems in language education include: they can be used regardless of time, i.e., are more available for learners, they can be easily integrated into chat-based applications that many people are familiar with, i.e., are more user-friendly, and they can be adapted to each learner using various information from chit-chat, i.e., are more supportive.

Needless to say, experiencing a substantial amount of production is critical in language acquisition. Nagata et al. (2020) showed that even a very primitive rule-based chatbot like ELIZA has the potential to increase learner’s sentence production. Their experiments also revealed that learners adopted words that appeared in the chatbot’s responses, suggesting that the expressions used by the dialog system had a positive impact on learners and that the system was effective in helping them learn unfamiliar words.

Considering these results, we propose a task of generating a response including the specified grammatical items. Here, grammatical items refer to such as to the present perfect, subjunctive, and relative clauses. Usually, they are gradually covered in a language learning course, typically through a school curriculum. Such responses can naturally expose learners to a variety of uses of a specific item and can give them experience of how to use the item in a variety of topics and situations, based on their own past experiences evoked in the conversation. In turn, we expect the learners to use the exposed constructions in their own production more as the exposed uses are linked tightly to their memories by encountering usage examples through dialog based on their own experiences.

The proposed task is formalized as follows. Given  $C = [c_1, c_2, \dots, c_n]$ , a dialog context that is a sequence of  $n$  utterances between two interlocutors (the system and the learner), and  $I$ , a set of grammatical items specified to be included, the task is to generate  $r$ , a natural response that follows  $c_n$ , on the condition that  $r$  includes an expression corresponding to each item  $i \in I$ . To the best of our knowledge, this is the first work to tackle this generation task for language learning.

To generate text that satisfies particular conditions, Lin et al. (2021) propose using auxiliary modules to guide pre-trained language models. Keskar et al. (2019) propose training language models with control code. Since these methods are based on supervised learning, they require annotated datasets. However, there is a lack of large labeled dialog datasets for grammatical items.

In this paper, we examine two approaches for generating responses containing the specified grammatical items without a large labeled dataset: 1) RL-based generation: fine-tuning a pre-trained language model using reinforcement learning (RL), and 2) Prompt-based generation: providing a large language model with prompt text with a task in-



struction and a few examples. The experiments confirm both approaches are promising.

## 2 Related Work

### 2.1 Dialog systems and adaptation in language learning

According to [Xiao et al. \(2023\)](#), there are three main uses for dialog systems in language learning.

One way is language learning through general communication. As one of the educational applications of dialog systems, there is a growing body of research on introducing dialog systems in second language learning through free interaction with dialog systems. Alexa ([Moussalli and Cardoso, 2020](#); [Dizon, 2017](#); [Dizon and Tang, 2020](#)) and Google Assistant ([Tai, 2022](#)) were used. In most studies, learners favorably accepted the system as a dialog partner.

Another way is task-based language learning. The introduction of a dialog system into a task allows for more content-focused learning. Tasks can be varied, such as asking for the time of day at a particular location or ordering at a coffee shop ([Wu et al., 2020](#); [Timpe-Laughlin et al., 2020](#)). Learners are allowed to interact and receive feedback throughout the task, which contributes to second language acquisition.

The third way is language learning based on structured pre-programmed dialog. To create a dialog on a specific topic, researchers design their system, rather than adapting a general dialog system. Many studies have been conducted with children. Some had three to six-year-olds learn to read through questions ([Xu et al., 2021a,b](#)), and had nine-year-olds answer their questions ([Lee and Jeon, 2022](#)). Another related survey is ([Huang et al., 2022](#)).

A further related area is user adaptation to difficulty in language tutoring. [Pandiarova et al. \(2019\)](#) worked on predicting the difficulty of fill-in-the-blank questions in which the words to be entered were specified.

Our study proposes a new task not addressed in these studies and provides new insights into methods for this task.

### 2.2 Reinforcement Learning

Reinforcement learning is a machine learning framework that acquires an optimal action policy based on non-instantaneous evaluations given by a reward function for a set of actions. By considering

the output tokens as actions, language generation can be treated as a reinforcement learning problem. Given an appropriate reward function, policy gradient methods such as REINFORCE ([Williams, 1992](#)) can fine-tune a pre-trained generative neural language model without a training dataset. In this paper, we adopt self-critical sequence training (SCST) ([Rennie et al., 2017](#)). SCST is proposed for image caption generation and is known for its simplicity and effectiveness.

The design of the reward function varies from task to task, but unlike the loss function in supervised learning, it allows the use of non-differentiable functions including the evaluation metrics used in text generation tasks such as BLEU and ROUGE ([Paulus et al., 2018](#); [Wu et al., 2018](#); [Narasimhan et al., 2016](#)).

Language generation based on deep learning generally uses cross-entropy as the loss function, which means that the objective function and the evaluation measure will be different. By incorporating the evaluation measure in the reward function, the gap can be alleviated.

### 2.3 Large-scale Pre-trained Language Model

In recent years, many researchers have studied methods for controlling the output of generative language models by providing prompts containing task instructions and examples as input ([Li et al., 2022](#); [Reynolds and McDonell, 2021](#); [Dou et al., 2022](#)).

In particular, GPT-3 ([Brown et al., 2020](#)) has achieved significant performance comparable to or better than other fine-tuned models in CoQA and TriviaQA in few-shot settings.

## 3 Method

For the sake of simplicity, in this paper, we assume context  $C$  contains only the immediately previous utterance ( $n = 1$ ). We also limit the number of specified items to 1 ( $|I| = 1$ ).

### 3.1 RL-based generation

For simplicity again, we train a different model for each grammatical item. In applications, we assume the models are to be switched given a learner's need. For example, when a learning partner chatbot finds that the learner tends to make errors with a particular item, the chatbot can increase the frequency of opting the generation model for the item than the vanilla generation model.

We consider three sub-functions for the reward,  $R_g$  for inclusion of grammatical items, which is the main objective,  $R_d$  for greater diversity, and  $R_f$  for higher fluency. The latter two are to mitigate learning bias towards including grammatical items. When only  $R_g$  is used, the model easily starts to exploit a fixed utterance against any input context. We will examine several combinations of these functions in our experiment in the next section.

**Reward on grammatical items** Let  $F_i(s) \in [0, 1]$  be a soft classifier that evaluates whether a given sentence  $s$  contains a specified grammatical item  $i$ . When we train a response generation model for item  $i$ , we set  $R_g(s) = F_i(s)$ .

For  $F_i(s)$ , we use BERT (Devlin et al., 2019). We obtain hidden representation  $\mathbf{h}_{[\text{CLS}]}$  of the [CLS] token from the final layer of a pre-trained BERT model.  $F_i(s)$  is formulated as follows:  $F_i(s) = \sigma(\mathbf{w}^\top \mathbf{h}_{[\text{CLS}]} + b)$ , where  $\sigma(\cdot)$  is the sigmoid function and  $\mathbf{w}, b$  are the learnable parameters. In training, the BERT model is not frozen and fine-tuned together with the parameters.

Although  $F_i(s)$  is trained in a supervised manner, the necessary data for this training is much more affordable than that for training a generation model. We will revisit this point in the next section.

**Rewards on diversity and fluency** We use Distinct-N (Li et al., 2016), an n-gram based diversity metric, as  $R_d$ . As  $R_f$ , we use the likelihood of the output  $r$  conditioned on the input, i.e., the dialog context  $C$ . The likelihood is computed by a pre-trained dialog model.

### 3.2 Prompt-based generation

In the same way with the RL-based approach, we prepare a prompt template for each item  $i$ . The templates are to be switched by applications.

Figure 1 shows a prompt template used in this study, which consists of an instruction indicating what the task is, some examples (called shots) and a query at the end.  $\langle c \rangle$  in Figure 1 is replaced with an input context utterance. Given an input prompt, a left-to-right generative language model outputs a sentence  $r$  that follows the prompt.

## 4 Experiment

We verified the effectiveness of both RL-based and prompt-based approaches using three items in the SCoRE corpus (Chujo et al., 2015): the present perfect, relational clause, and subjunctive.

---

A and B are speaking. Create B’s response using the present perfect.  
 ===  
 A: Good morning, how are you doing today?  
 B: I have been feeling pretty good, Dr. Smith.  
 ===  
 A: What’s your plan for your future?  
 B: I’d like to work in a law firm to enrich my experience and put what I’ve learned into practice.  
 ===  
 A: I’m going to Japan this year on vacation.  
 B: Have you ever been to America?  
 ===  
 A:  $\langle c \rangle$   
 B:

---

Figure 1: Prompt template for the present perfect tense

### 4.1 Datasets

In accordance with the assumption of  $n = 1$ , we extracted only the first utterance pair of each dialog from the Daily Dialog corpus<sup>1</sup> (Li et al., 2017) to compose our dataset. The first utterance of each pair was used as a context  $C$ , and the second was used as a reference (used for analysis purposes). We split the pairs into three subsets: 10,618 for training, 500 for development, and 1,000 for test.

We used the SCoRE corpus to build  $F_i(\cdot)$ . We built a classifier for each of the three items above. Appendix A gives the details of the SCoRE dataset, classifier training, and performance. Note that the required data for training here need not be dialog data and can be much smaller than that for supervised training of a dialog language model.

### 4.2 Evaluation metrics

We used three metrics for our evaluation. First, we defined the function  $\delta_i(s)$ , which returns 1 or 0 for sentence  $s$  by using  $F_i(s)$  with a threshold of 0.5.

As the first metric, we introduced G-ratio to measure the capability of the model to generate responses that include the specified grammatical item. G-Ratio indicates the percentage of outputs containing the item and can be automatically measured by using  $\delta_i(s)$ .

Considering our aim of exposing learners to various uses of grammatical items in dialog, the model should be able to return diverse responses. We adopted Distinct-N (N=2) as the second metric.

Finally, we defined GOAL (Grammar Oriented Average Likelihood), which measures the fluency of only the generated sentences that contain the specified item using the output likelihood based on

<sup>1</sup>[https://huggingface.co/datasets/daily\\_dialog](https://huggingface.co/datasets/daily_dialog)

a dialog language model  $P_m$  as follows:

$$H_i^T = \{s \in G_i^T \mid \delta(s) = 1\},$$

$$\text{GOAL}(H_i^T; P_m) = \frac{\sum_{s \in H_i^T} P_m(s \mid c(s))}{|H_i^T|},$$

where  $G_i^T$  the set of the generated responses given test set  $T$  in terms of item  $i$ , and  $H_i^T$  is the set of responses in  $G_i^T$  that  $F_i(\cdot)$  evaluated as containing the grammatical item.  $c(s)$  denotes the input context for output  $s$ .

### 4.3 Experimental setups

For the RL-based approach, we used DialoGPT (Zhang et al., 2020), a GPT-2 based dialog language model trained on a Reddit corpus, as the initial model in SCST, the main body of  $R_f$ , and  $P_m$ . For decoding, we used top- $k$  sampling (Fan et al., 2018) ( $k = 50$ ). The model was evaluated every 10 batches using the development data, and training was stopped with a patience of 3. As training progressed, the number of sentences containing the target grammatical item increased, but many similar sentences were generated, resulting in a loss of diversity. Therefore, as we observed a trade-off between G-Ratio and diversity, we adopted the product of the two as an indicator of early stopping.

For the prompt-based approach, we used GPT-3 davinci. We set the sampling temperature to 1 for GPT-3. Other settings are detailed in Appendix B.

### 4.4 Evaluation

For the RL-based approach, ten sentences were generated using beam search with a beam width of 10 for each test case. Out of the ten, the sentence with the highest likelihood and the specified item is chosen as the output. If no sentence included the item, the first one was chosen. We compare the following five combinations of the reward functions:  $R_g$ ,  $R_g + R_d$ ,  $R_g \times R_d$ ,  $R_g + R_f$ , and  $R_g \times R_f$ .

For the prompt-based approach, ten sentences were generated thorough the web API using a prompt for each test case, from which one was picked as above. We compared the following five variations, which combines 0, 1, and 3 task examples (called shots) and with/without task instructions: instr., 1-shot, 3-shots, instr.+1-shot, and instr.+3-shots. For example, ‘‘instr.’’ means 0-shot with instructions. ‘‘1-shot’’ means 1-shot without instructions. ‘‘instr.+3-shot’’ means 3-shot with instructions.

All metrics were applied to 1,000 outputs.

## 5 Results

Table 1 shows the results for each grammatical item. Example outputs are shown in Appendix C.

$R_g \times R_f$  showed the highest GOAL for the present perfect and the subjunctive, while  $R_g + R_f$  showed the highest GOAL for the relative clause.

The RL-based approach successfully improved G-Ratio in all cases. Although the Dist.-2 values got lower than before training (Baseline), this was expected in advance as the result of introducing a grammatical constraint in generation.

In the RL-based approach, a higher Dist.-2 tended to be obtained with the fluency reward function  $R_f$  than with the diversity reward function  $R_d$  except for the subjunctive, suggesting that the effect of  $R_d$  was limited. The reasons for this may be as follows. Even if sentences with a high Dist.-2 are more likely to be generated, it does not necessarily reflect the diversity of the model overall, and if the input sentences in the batch are similar, Dist.-2 in the output will naturally decrease, but the current reward function does not fully take this into account. In addition, taking fluency into account suppresses the abuse of fixed patterns (fixed patterns increase  $R_g$  but decrease diversity). For all grammatical items tested, GOAL improved when the reward function for fluency,  $R_f$ , was applied.

In the prompt-based approach, G-Ratio tended to be higher for inputs with both task instruction and shots. However, 3-shots sometimes gave worse results than 1-shot. This suggests that task instructions should be included in the input, but that increasing the number of shots may add noise or unintended bias to the language model, making it more difficult to obtain the desired output.

Comparing the two approaches, the prompt-based one demonstrated higher diversity than the RL-based one, and a comparable G-Ratio. Though the GOAL scores for the RL-based approach were higher than those for the prompt-based approach, we must note that GOAL is favorable to the RL-based approach that, in this paper, uses the same DialoGPT model as GOAL. As far as we manually compared the concrete responses from GPT-3 and DialoGPT for a small number of randomly picked cases, we did not find significant differences.

## 6 Discussion

Even though we want to expose more instances of a particular item to a learner, it is not natural to include the item in every dialog response. Therefore,

Table 1: Generation results of plain DialoGPT, DialoGPT fine-tuned by RL, and GPT-3 with prompts.

Model	Method	Present perfect			Relative clause			Subjunctive		
		G-Ratio	Dist.-2	GOAL	G-Ratio	Dist.-2	GOAL	G-Ratio	Dist.-2	GOAL
DialoGPT	Baseline	0.145	0.588	0.096	0.822	0.426	0.103	0.037	0.755	0.084
DialoGPT (RL)	w/ $R_g$	0.789	0.264	0.088	0.911	0.388	0.124	0.860	0.197	0.114
	w/ $R_g + R_d$	0.781	0.121	0.120	0.888	0.355	0.119	0.566	0.182	0.101
	w/ $R_g \times R_d$	0.789	0.265	0.093	0.854	0.411	0.096	0.794	0.207	0.091
	w/ $R_g + R_f$	0.792	0.290	0.107	0.896	0.386	<b>0.139</b>	0.941	0.095	0.214
	w/ $R_g \times R_f$	0.603	0.186	<b>0.147</b>	0.833	0.420	0.110	0.949	0.036	<b>0.241</b>
GPT-3 (prompt)	w/ instr.	0.735	0.681	0.014	0.996	0.682	0.017	0.279	0.737	0.014
	w/ 1-shot	0.493	0.701	0.016	0.992	0.575	0.041	0.568	0.512	0.036
	w/ 3-shots	0.514	0.666	0.027	0.997	0.563	0.038	0.359	0.593	0.033
	w/ instr. + 1-shot	0.901	0.511	0.035	0.997	0.588	0.034	0.721	0.484	0.031
	w/ instr. + 3-shots	0.753	0.594	0.033	0.997	0.571	0.036	0.535	0.539	0.031

we do not need to pursue 100% for G-Ratio.

We presented GOAL as a primary metric candidate for the proposed task. However, as noted in the previous section, it is not reliable when one wants to compare two results based on different language models. Taking the similarity to the reference sentences into account is one direction to mitigate this issue. Another strategy is combining GOAL with reference-free unsupervised dialog evaluation methods using follow-ups such as FULL (De Bruyn et al., 2022). Unlike GOAL, these evaluation methods do not measure the likelihood of the target utterances directly; they, however, still rely on a particular language model. A simple way to make this issue easier would be an ensemble approach using multiple language models or majority voting.

Considering the high diversity and the nature of training-free, so far the prompt-based approach seems to be advantageous, assuming the availability of a huge pre-trained model such as GPT-3. However, the RL-based approach may have merits in terms of its fine-grained, delicate, and implicit control than the prompt-based approach. (Besides, DialoGPT and GPT-2 did not work in the prompt-based approach. See Appendix C.)

## 7 Conclusion

We have proposed a new task of generating a response including the specified grammatical items for language learners. We examined two approaches and found that both are feasible.

Future directions include the expansion of the grammatical items. To push this task to practical use, locating appropriate places in conversations to include the items is also important.

This paper aimed to increase learners’ exposure to specific grammatical items, but another inter-

esting direction is generating preceding utterances that encourage or facilitate learners to use specific grammatical items in their next utterances.

## 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.
- Kiyomi Chujo, Kathryn Oghigian, and Shiro Akasegawa. 2015. A corpus and grammatical browsing system for remedial EFL learners. *Multiple affordances of language corpora for data-driven learning*, pages 109–130.
- Maxime De Bruyn, Ehsan Lotfi, Jeska Buhmann, and Walter Daelemans. 2022. [Open-domain dialog evaluation using follow-ups likelihood](#). In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 496–504, Gyeongju, Republic of Korea. International Committee on 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.
- Gilbert Dizon. 2017. [Using intelligent personal assis-](#)

- tants for second language learning: A case study of alexa. *TESOL Journal*, 8:811–830.
- Gilbert Dizon and Daniel Tang. 2020. [Intelligent personal assistants for autonomous second language learning: An investigation of alexa](#). *The JALT CALL Journal*, 16.
- Yao Dou, Maxwell Forbes, Rik Koncel-Kedziorski, Noah A Smith, and Yejin Choi. 2022. Is gpt-3 text indistinguishable from human text? scarecrow: A framework for scrutinizing machine text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7250–7274.
- 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.
- Weijiao Huang, Khe Hew, and Luke Fryer. 2022. [Chatbots for language learning-are they really useful? a systematic review of chatbot-supported language learning](#). *Journal of Computer Assisted Learning*.
- Yasutake Ishii and Yukio Tono. 2018. Investigating japanese efl learners’ overuse/underuse of english grammar categories and their relevance to cefr levels. In *Proceedings of the 4th Asia Pacific Corpus Linguistics Conference*, pages 160–165.
- Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.
- Na-Young Kim. 2016. Effects of voice chat on efl learners’ speaking ability according to proficiency levels. *Multimedia-Assisted Language Learning*, 19(4):63–88.
- Seongyong Lee and Jaeho Jeon. 2022. [Visualizing a disembodied agent: young efl learners’ perceptions of voice-controlled conversational agents as language partners](#). *Computer Assisted Language Learning*, 0(0):1–26.
- Jiaoda Li, Ryan Cotterell, and Mrinmaya Sachan. 2022. Probing via prompting. *arXiv preprint arXiv:2207.01736*.
- 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.
- Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957*.
- Zhaojiang Lin, Andrea Madotto, Yejin Bang, and Pascale Fung. 2021. The adapter-bot: All-in-one controllable conversational model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 16081–16083.
- Souheila Moussalli and Walcir Cardoso. 2020. [Intelligent personal assistants: can they understand and be understood by accented l2 learners?](#) *Computer Assisted Language Learning*, 33(8):865–890.
- Ryo Nagata, Tomoya Hashiguchi, and Driss Sadoun. 2020. Is the simplest chatbot effective in english writing learning assistance? In *Computational Linguistics*, pages 245–256, Singapore. Springer Singapore.
- Karthik Narasimhan, Adam Yala, and Regina Barzilay. 2016. [Improving information extraction by acquiring external evidence with reinforcement learning](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2355–2365, Austin, Texas. Association for Computational Linguistics.
- Irina Pandarova, Torben Schmidt, Johannes Hartig, Ahcene Boubekki, Roger Jones, and Ulf Brefeld. 2019. [Predicting the difficulty of exercise items for dynamic difficulty adaptation in adaptive language tutoring](#). *International Journal of Artificial Intelligence in Education*.
- Romain Paulus, Caiming Xiong, and Richard Socher. 2018. [A deep reinforced model for abstractive summarization](#). In *International Conference on Learning Representations*.
- Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7008–7024.
- Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–7.
- Sherry Ruan, Angelica Willis, Qian Yao Xu, Glenn M Davis, Liwei Jiang, Emma Brunskill, and James A Landay. 2019. Bookbuddy: Turning digital materials into interactive foreign language lessons through a voice chatbot. In *Proceedings of the Sixth (2019) ACM Conference on Learning@ Scale*, pages 1–4.
- Tzu-Yu Tai. 2022. [Effects of intelligent personal assistants on efl learners’ oral proficiency outside the classroom](#). *Computer Assisted Language Learning*, 0(0):1–30.
- Stergios Tegos, Stavros Demetriadis, and Thrasylvoulos Tsiatsos. 2014. A configurable conversational agent to trigger students’ productive dialogue: a pilot study in the call domain. *International Journal of Artificial Intelligence in Education*, 24(1):62–91.

- Veronika Timpe-Laughlin, Tetyana Sydorenko, and Phoebe Daurio. 2020. [Using spoken dialogue technology for L2 speaking practice: what do teachers think?](#) *Computer Assisted Language Learning*, 35:1–24.
- Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256.
- Lijun Wu, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018. [A study of reinforcement learning for neural machine translation](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3612–3621, Brussels, Belgium. Association for Computational Linguistics.
- Yunhan Wu, Daniel Rough, Anna Bleakley, Justin Edwards, Orla Cooney, Philip R. Doyle, Leigh Clark, and Benjamin R. Cowan. 2020. [See what i'm saying? comparing intelligent personal assistant use for native and non-native language speakers](#). In *22nd International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI '20*, New York, NY, USA. Association for Computing Machinery.
- Feiwen Xiao, Priscilla Zhao, Hanyue Sha, Dandan Yang, and Mark Warschauer. 2023. [Conversational agents in language learning](#). *Journal of China Computer-Assisted Language Learning*.
- Ying Xu, Joseph Aubele, Valery Vigil, Andres Bustamante, Young-Suk Kim, and Mark Warschauer. 2021a. [Dialogue with a conversational agent promotes children's story comprehension via enhancing engagement](#). *Child Development*, 93.
- Ying Xu, Dakuo Wang, Penelope Collins, Hyelim Lee, and Mark Warschauer. 2021b. [Same benefits, different communication patterns: Comparing children's reading with a conversational agent vs. a human partner](#). *Computers & Education*, 161:104059.
- Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. [DIALOGPT : Large-scale generative pre-training for conversational response generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

## Supportive Material (Appendices)

### A Classifier for grammatical items

We used a classifier that determines whether a grammatical item is included or not as a reward function for RL. The structure of the classifier is as described in §3.1, where the input sentences to be judged are estimated to determine whether they contain grammatical items or not by a linear layer and a sigmoid function based on the embedding of BERT’s [CLS] tokens.

The classifier requires a dataset for training. However, the required data need not be interactive, and can be smaller than for supervised learning of a language model. When data is not available, regular expression-based classification can be used as a substitute.

In this section, we describe the dataset used to train the classifier and the settings. The performance of the classifier is compared with rule-based classification using regular expressions. The regular expressions were created on the basis of the CEFR-J regular expression list (Ishii and Tono, 2018).

#### A.1 SCoRE Corpus

The SCoRE corpus, in which grammatical items are manually assigned to sentences, was used to train the classifier. Therefore, the grammatical items were those included in the SCoRE corpus. The SCoRE corpus contains approximately 20 grammatical items, and Table 2 shows the number of data corresponding to the grammatical items used in this study. For example, in the subjunctive, I wish, if I were, if + verb past tense, if + had + verb past participle, etc. are included in the data.

In addition to positive examples with the target grammar item, negative examples without the item are required to train the classifier. Therefore, for the negative examples, we use sentences in the SCoRE corpus that are assigned grammatical items that are not the target ones. However, if all sentences that do not have the target grammar item are used as negative examples, there is a possibility that unsuitable data will be included, and the proportion of unsuitable data will be greatly biased. We constructed a dataset for training by extracting data from the negative examples in the dataset in such a way that there is no bias in the number of positive examples.

From the data obtained, 80% was split into training data and 20% into test data. Finally, for the

Table 2: SCoRE dataset statistics

Grammatical item	# of text
Present Perfect	547
Relative Clause	1,142
Subjunctive	783

Table 3: Results: Classifier Accuracy

Grammatical item	BERT	Regular expression
Present Perfect	0.9902	0.9641
Relative Clause	0.9879	0.6909
Subjunctive	0.9919	0.7394

present perfect, the training data and test data were 1,222 and 306, respectively, and for the relational clause and hypothetical, the training data and test data were 1,977 and 495, respectively.

#### A.2 Hyperparameter for training the classifier

We used BERT (bert-large-uncased) to set the initial values for the classification model. Parameters were optimized by AdamW during training. The learning rate was set to  $2e^{-5}$  and the coefficient of L2 regularization to  $1e^{-2}$ . The batch size was set to 10 and the number of epochs was set to 10. In this experiment, the classifier is the model that performed best on the test data.

#### A.3 Classification Performance

Table 3 shows the classification performance of the classifiers for each grammar item. The evaluation was conducted using the percentage of correct answers between the correct and predicted labels as the evaluation measure. In the experiment, the BERT-based classifier was used as the reward function for the other items because BERT had better classification performance than the regular expression.

### B Hyperparameter in the experiment

In top- $k$  sampling in SCST, we set  $k$  to 50. For Distinct-N in  $R_d$ ,  $N = 2$ . The parameters were optimized by AdamW during training, with a learning rate of  $2e^{-5}$  and a coefficient of L2 regularization of  $1e^{-2}$ . The minimum output length was set to 10 in order to properly compute Distinct-N. The batch size was set to 10, with a maximum of 1100 iterations. For GPT-3, we set engine to davinci, max\_tokens to 20, temperature to 1, n to 10, and stop to "\n".

Table 4: Generation results for DialoGPT, GPT-2, and GPT-3 with prompts.

Model	Method	Present perfect			Relative clause			Subjunctive		
		G-Ratio	Dist.-2	GOAL	G-Ratio	Dist.-2	GOAL	G-Ratio	Dist.-2	GOAL
DialoGPT	w/ instr. + 1-shot	0.065	0.182	0.108	0.953	0.096	0.091	0.235	0.073	0.081
	w/ instr. + 3-shots	0.569	0.051	0.040	0.960	0.237	0.013	0.049	0.292	0.026
GPT-2	w/ instr. + 1-shot	0.753	0.131	0.012	0.943	0.191	0.029	0.201	0.163	0.007
	w/ instr. + 3-shots	0.638	0.071	0.015	0.955	0.276	0.022	0.253	0.211	0.008
GPT-3	w/ instr. + 1-shot	0.901	0.511	0.035	0.997	0.588	0.034	0.721	0.484	0.031
	w/ instr. + 3-shots	0.753	0.594	0.033	0.997	0.571	0.036	0.535	0.539	0.031

## C Examples

In this section, we provide generated sentences of compared methods. First, we discuss additional smaller models we experimented with in addition to the GPT-3. Next, we show samples of outputs for two inputs for several RL-based and prompt-based methods.

### C.1 Other Models in the Prompt-based Approach

We also tested the performance of GPT-2 and DialoGPT in the same settings as GPT-3. Table 4 shows the results. Comparing the performance of the three models in terms of G-Ratio, GPT-3, which has the largest model size, shows the best performance, while GPT-2 tends to perform better than DialoGPT. In GOAL, GPT-3 showed consistently high, but DialoGPT also showed high values in some settings. Note, however, that DialoGPT was used in the GOAL calculations and is a favorable indicator for this model. Also, GPT-2 and DialoGPT did not seem to produce higher quality responses than GPT-3, as far as we could visually confirm. (See Appendix C.2) Therefore, GPT-3 is superior to the other models in terms of both the G-Ratio and GOAL value, regardless of the grammatical items, and in terms of the quality of the response sentences.

### C.2 Samples

Table 5, 6 show examples of output in the present perfect tense with different input contexts. Compared with the Daily Dialog corpus and DialoGPT, after learning, the response sentences are in the present perfect tense, and the responses of the method that performed well in our experiments are not too broken to be used as a dialog response. However, some of the methods showed unstable output, such as repetition of similar sentences or very few words.



Input context	Look at the show on TV. I am watching a food show at a very famous seafood restaurant. I really want to eat at that restaurant. I am a seafood lover.
Daily dialog (reference)	Speaking of seafood , my mouth is watering. Let's go to the seafood restaurant in our neighborhood.
DialoGPT	I love seafood!
DialoGPT w/ $R_g + R_d$	I've <b>been</b> there. I've <b>been</b> there. I've <b>been</b> there. I've <b>been</b> there. I've <b>been</b> there. ... I've <b>been</b> there. I've <b>been</b> there. I've <b>been</b> there. I've <b>been</b> there. I've <b>been</b> there. ... I've <b>been</b> there. I've <b>been</b> there. I've <b>been</b> there. I've <b>been</b> there! I've <b>been</b> there! ... I've <b>been</b> there. I've <b>been</b> there. I've <b>been</b> there. I've <b>been</b> there. I've <b>been</b> there! ... I've <b>been</b> there. I've <b>been</b> there. I've <b>been</b> there. I've <b>been</b> there. I've <b>been</b> there. ...
DialoGPT w/ $R_g \times R_d$	I've <b>never been</b> to a seafood restaurant, but I've <b>heard</b> good things! I've <b>never been</b> to a seafood restaurant, but I've <b>always wanted</b> to go to one. I've <b>never been</b> to a seafood restaurant, but I've <b>always wanted</b> to. I've <b>never been</b> to a seafood restaurant, but I've <b>always wanted</b> to try one. I've <b>never been</b> to a seafood restaurant, but I've <b>heard</b> good things about them.
DialoGPT w/ $R_g + R_f$	I've <b>never been</b> to a seafood restaurant, but I've <b>heard</b> good things. I've <b>never been</b> to a seafood restaurant, but I've <b>heard</b> good things about them. I've <b>never been</b> to a seafood restaurant, but I've <b>always wanted</b> to go there. I've <b>never been</b> to a seafood restaurant, but I've <b>always wanted</b> to try one. I've <b>never been</b> to a seafood restaurant, but I've <b>always wanted</b> to go to one.
DialoGPT w/ $R_g \times R_f$	I've <b>never been</b> to a seafood restaurant. I love seafood. I want to go to that restaurant. I've <b>never been</b> to a seafood restaurant, but I've always wanted to. I've <b>never been</b> to a seafood restaurant, but I've heard good things.
DialoGPT w/ instr. + three-shots	I That You Weird H
GPT-2 w/ instr. + three-shots	I'm going to Japan this year. Do you have any plans to go to Japan this year? I want to eat at that restaurant. Do you have a plan for your future? I want to eat at that restaurant.
GPT-3 w/ instr.+ one-shots	I think it is kind of expensive to eat at that restaurant . I want to go to that I see, the seafood restaurant's food is always delicious here. I am very hungry right now I <b>have never heard</b> of that restaurant . It must be very famous . Yes, that is a very famous food and show but I Love shark fin soup I kind of miss home cooking . I really want to make some of my favorite easy dishes.
GPT-3 w/ instr.+three-shots	I've <b>never been</b> to that restaurant for a lunch buffet . I <b>have been</b> wanting to have sea food, too . I will take the test . Is there any part that you want to eat first ? I want to go to Japan someday. That sounds really delicious . I love ALL foods, but I like seafood too .

Table 5: Output samples of considered methods and DialoGPT. (1)

Input context	Do you have a package tour to Hollywood?
Daily dialog (reference)	Yes, here is the table, which plans do you like best?
DialoGPT	I don't have a package tour.
DialoGPT w/ $R_g + R_d$	I've <b>been</b> there, done that, and done that. I've <b>been</b> there, done that, and done that. ... I've <b>been</b> there, done that, and done that. I've <b>been</b> there, done that. I've <b>been</b> there, ... I've <b>been</b> there, done that, and done that. I've <b>been</b> there, done that. I've <b>been</b> there, ... I've <b>been</b> there, done that, and done that. I've <b>been</b> there, done that, and done that. ... I've <b>been</b> there, done that, and done that. I've <b>been</b> there, done that. I've <b>been</b> there, ...
DialoGPT w/ $R_g \times R_d$	No, I've <b>never been</b> to Hollywood. I've <b>never been</b> to Hollywood, but I've <b>heard</b> good things about it. Not yet, but I've <b>been</b> thinking about it. I've <b>never been</b> to Hollywood, but I've <b>heard</b> good things. I've <b>never been</b> to Hollywood, but I've <b>heard</b> it's great!
DialoGPT w/ $R_g + R_f$	No, I've <b>never been</b> to Hollywood. Not yet, but I've <b>been</b> thinking about it. No, I've <b>never been</b> to Hollywood before. Not yet, but I've <b>been</b> thinking about it! I've <b>never been</b> to Hollywood, but I've <b>heard</b> good things about it.
DialoGPT w/ $R_g \times R_f$	I don't, but I've <b>heard</b> of it. No, I've <b>never been</b> to Hollywood. No, but I've <b>been</b> thinking about it. No, but I've <b>been</b> thinking about doing one. No, I've <b>never been</b> to Hollywood, but I've heard good things about it.
DialoGPT w/ instr. + three-shots	<b>I have not</b> <b>I 've never</b> I don't Haha, I'd like
GPT-2 w/ instr. + three-shots	I'd like to work in a law firm to enrich my experience and put what I've <b>learned</b> into practice. I don't have a package tour. Yes. I would love to. I don't have a package tour to Hollywood.
GPT-3 w/ instr.+ one-shots	No , I don't. However , I do have individual support to resit your broken leg I don't have any right now , I just got back yesterday from a vacation in Hawaii . After taking several Korean movies , I decided I didn't need to go . No , but I just returned from my vacation to Hollywood and Yellowstone Park last night. Yes , I do . And I completely planned on it ,too.
GPT-3 w/ instr. + three-shots	Definitely . In fact, I <b>have been</b> living in Hollywood for around twenty days . I'm not sure, but I will contact our office about it. Yes , I do. Would you like to book? Traveling by yourself is more fun than traveling in a group . No, but we have a tour to San Francisco .

Table 6: Output samples of the considered methods and DialoGPT. (2)

# UKP-SQuARE: An Interactive Tool for Teaching Question Answering

Haishuo Fang, Haritz Puerto, Iryna Gurevych

Ubiquitous Knowledge Processing Lab (UKP Lab),  
Department of Computer Science and Hessian Center for AI (hessian.AI),  
Technical University of Darmstadt  
[www.ukp.tu-darmstadt.de](http://www.ukp.tu-darmstadt.de)

## Abstract

The exponential growth of question answering (QA) has made it an indispensable topic in any Natural Language Processing (NLP) course. Additionally, the breadth of QA derived from this exponential growth makes it an ideal scenario for teaching related NLP topics such as information retrieval, explainability, and adversarial attacks among others. In this paper, we introduce UKP-SQuARE as a platform for QA education. This platform provides an interactive environment where students can run, compare, and analyze various QA models from different perspectives, such as general behavior, explainability, and robustness. Therefore, students can get a first-hand experience in different QA techniques during the class. Thanks to this, we propose a learner-centered approach for QA education in which students proactively learn theoretical concepts and acquire problem-solving skills through interactive exploration, experimentation, and practical assignments, rather than solely relying on traditional lectures. To evaluate the effectiveness of UKP-SQuARE in teaching scenarios, we adopted it in a postgraduate NLP course and surveyed the students after the course. Their positive feedback shows the platform’s effectiveness in their course and invites a wider adoption.

## 1 Introduction

Question Answering (QA) is one of the overarching research topics in Natural Language Processing (NLP). QA pipelines have been developed to address different types of questions, knowledge sources, and answer formats, including extractive, abstractive, knowledge base, multiple-choice, generative, and open-domain QA. Such a massive number of QA systems and relevant NLP techniques are making QA lectures more important in NLP courses. However, despite QA being an application-oriented topic (e.g., chatbots, virtual assistants, etc.), classes are usually theoretically

driven. Thus, in this paper, we propose the use of the UKP-SQuARE platform as a tool for QA education. This platform integrates most QA formats, popular models, datasets, and analysis tools, such as explainability, adversarial attacks, and graph visualizations.

Compared with conventional teacher-led classes, we propose a learner-centered class following the flipped classroom (Bishop and Verleger, 2013) with UKP-SQuARE as the driving tool of the lecture. This tool provides an interface for users to interact with different QA models and analysis tools. Therefore, students can actively learn about QA systems and get hands-on experience by interacting with models on the platform. Concretely, students can flexibly compare multiple architectures that model different QA formats, analyze their outputs with explainability tools, and even analyze their robustness against adversarial attacks. Prior studies have shown that flipped classroom lectures improve the learning process of students in programming courses (Alhazbi, 2016). Thus, we believe that teaching and learning QA through a live demo with this platform can also make NLP lectures more engaging, drawing students’ attention, and interest in the topics.

To investigate the effectiveness of UKP-SQuARE in QA education, we adopted it for the first time in a postgraduate NLP course<sup>1</sup> and conducted a survey afterward. The positive feedback from the students encourages us to continue adopting this platform and education method in more NLP courses. The contributions of this paper are: i) a novel interactive learner-centered methodology to teach QA and relevant NLP topics, ii) extending the UKP-SQuARE platform for teaching QA, and iii) the design of a syllabus for interactive QA lectures.

<sup>1</sup>Master’s level course

## 2 UKP-SQuARE

UKP-SQuARE (Baumgärtner et al., 2022; Sachdeva et al., 2022; Puerto et al., 2023) is an extendable and interactive QA platform that integrates numerous popular QA models such as deeppset’s roberta-base-squad<sup>2</sup>, SpanBERT (Joshi et al., 2020) for HotpotQA, and QAGNN (Yasunaga et al., 2021). It provides an ecosystem for QA research, including comparing different models, explaining model outputs, adversarial attacks, graph visualizations, behavioral tests, and multi-agent models. In addition, this platform provides a user-friendly interface<sup>3</sup> that enables users to interact. Users can run available models, deploy new ones, compare their behaviors, and explain outputs.

## 3 Learning Question Answering with UKP-SQuARE

In this section, we present the syllabus of a lecture focused on QA and relevant NLP topics that use the platform UKP-SQuARE following the flipped classroom methodology (Bishop and Verleger, 2013). The flipped classroom is an effective learner-centered educational methodology in which students study pre-recorded lectures and materials in advance to engage in more interactive and collaborative learning activities in class. UKP-SQuARE can be the driving tool for the flipped classroom in QA education. With our platform, lecturers can introduce the topics by interacting with the students and then proceed to an in-depth explanation of the technical details behind the methods of each topic. We propose dividing the lecture into three topics in the QA field: basic QA concepts, trustworthy QA, and multi-agent QA systems. With these topics, students can learn about QA and related NLP topics such as information extraction, explainability, adversarial attacks, and multi-agent systems.

### 3.1 Learning Basic QA Components

QA systems include two main components, i.e., Readers and Retrievers. Readers are QA models responsible for obtaining answers from the context retrieved by retrievers. In UKP-SQuARE, students can easily learn various readers (QA models) within different QA formats and information retrieval techniques via interacting with the interface.

<sup>2</sup><https://huggingface.co/deeppset/roberta-base-squad2>

<sup>3</sup><https://square.ukp-lab.de/>

### 3.1.1 Contrasting Different QA Formats

With UKP-SQuARE, students can get first-hand experience by interacting with multiple models on our platform. The home readings would include descriptions of the main QA datasets and their base-lines. In class, the lecturer can show the different QA formats with real demonstrations of the models and explain on the fly the architectural differences needed to model each QA format. An example is shown in Figure 1 where a span-extraction QA model, i.e., Span-BERT, and a multiple-choice QA model, i.e., CommonsenseQA model are presented to show the difference between these two QA formats. Such interactions can make theoretical explanations of the architectures easier to digest and, therefore, the class more engaging.

### 3.1.2 Learning Information Retrieval

To learn Information Retrieval (IR) methods, the user interface of UKP-SQuARE offers a compelling approach to help students differentiate between different IR methods, e.g., lexical retrieval and semantic retrieval, and understand how they affect the final performance of QA models. The home readings would include book chapters or slides describing the main IR methods such as TF-IDF (Sparck Jones, 1988), BM25 (Robertson et al., 1995), Sentence-BERT (Reimers and Gurevych, 2019), and Dense Passage Retrieval (DPR; Karpukhin et al., 2020). Like the above section, the lecturer can guide students to find the difference between lexical retrieval (e.g., BM25) and semantic retrieval (e.g., DPR) via playing with UKP-SQuARE by themselves. As shown in Figure 2, for the question *When was Barack Obama’s inauguration?*, the BM25 retriever returns a passage covering all keywords but irrelevant to the question, while the DPR retriever returns the correct document, which contains the answer to the question. By providing this example in class, students can easily understand that DPR retrieves semantically similar passages while BM25 only retrieves passages that contain the query tokens and, thus, may retrieve unrelated passages. This could be further explored by comparing two open-domain QA models implementing these retrieval methods and the same reader model to demonstrate the error propagation due to irrelevant passages. This active learning method can prevent the issue of students losing attention that commonly occurs in traditional lectures (Felder and Brent, 2003).

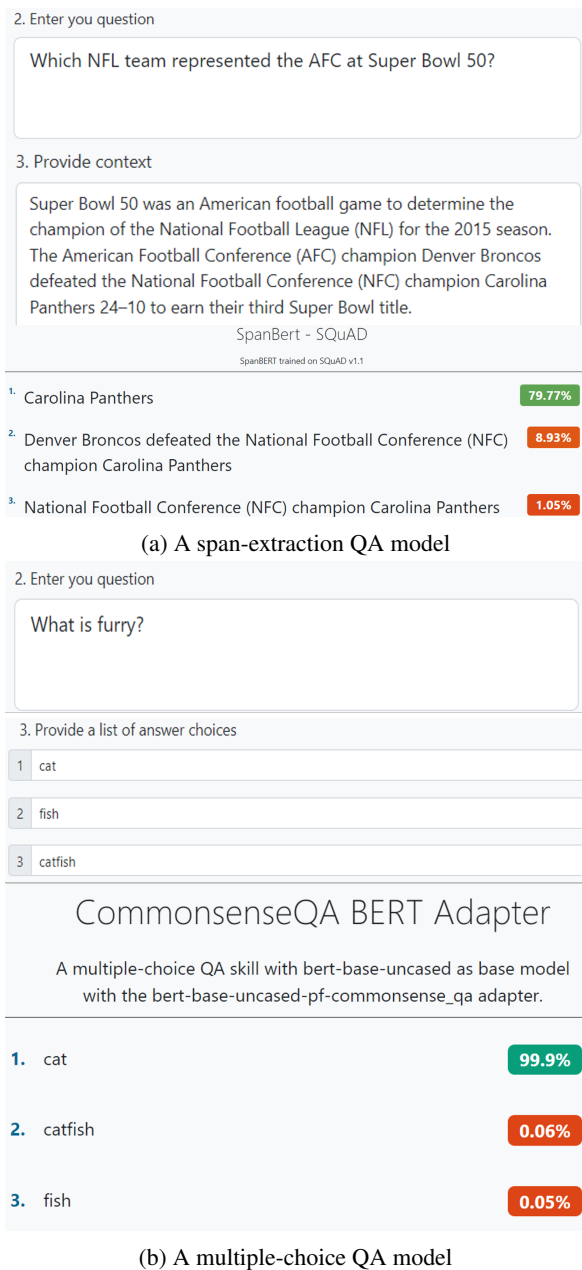


Figure 1: Different QA formats in UKP-SQuARE

### 3.2 Learning Trustworthy QA Systems

In addition to learning basic QA components, it is important to understand how to identify and evaluate trustworthy QA systems. This involves several related NLP topics, such as explainability, transparency, and robustness. UKP-SQuARE provides such analysis tools to facilitate students' learning process of trustworthy QA systems.

#### 3.2.1 Explainability Methods

The exponential adoption of AI is pushing regulators to adopt policies to regulate its use. One of the key points they aim to address is the explainabil-

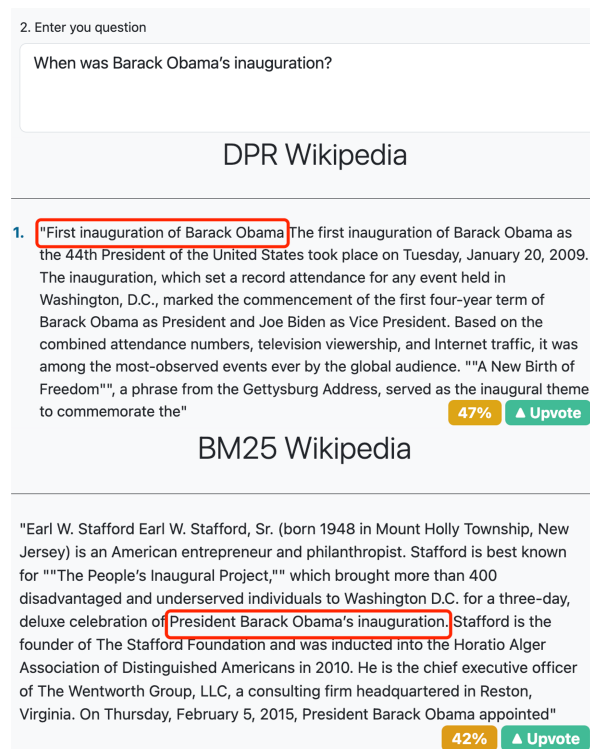


Figure 2: Example of difference between using BM25 retriever and DPR retriever. The red boxes represent keywords in the retrieved passages

ity of these methods to make AI safer<sup>4</sup>. Thus, it is of utmost importance to include explainability methods on AI courses in Universities. In terms of the explainability of QA models, UKP-SQuARE includes BertViz (Vig, 2019) and a suite of saliency map methods to facilitate the understanding of the model's decision-making process. Saliency maps employ attribution-weighting techniques such as gradient-based (Simonyan et al., 2014; Sundararajan et al., 2017) and attention-based (Jain et al., 2020; Serrano and Smith, 2019) methods to determine the relative importance of each token for the model prediction. The descriptions of these methods would form part of the home readings and to make the classes more active, the class would be driven by real examples of saliency maps using our platform and their interpretation. In this way, students can learn how to explain the output of a QA model based on saliency maps.

An example of a saliency map is shown in Figure 3. The color level of the highlighted text reflects its importance for the answer. As we can see, *of what celestial body?* is the most important part of

<sup>4</sup><https://digital-strategy.ec.europa.eu/en/policies/european-approach-artificial-intelligence>

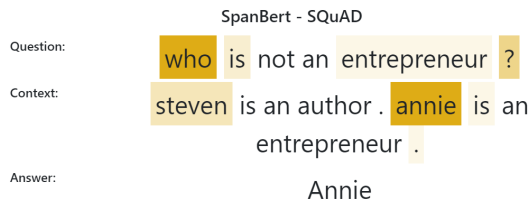


Figure 3: An attention-based saliency map of a question in UKP-SQuARE.

the question, while *sun* gets the most attention in the context, which is the final answer. This means the model successfully understands the main point of the question and can link them to the context. Making this type of interpretation can help students identify potential problems or biases in the models.

### 3.2.2 Behavioral Tests in QA models

The next important component in trustworthy QA is behavioral tests of models. Machine learning models do not throw errors as regular software programs. Instead, an error in machine learning is usually an unwanted behavior, such as a misclassification that may pass inadvertently to a person (Ribeiro et al., 2020). This makes testing machine learning models challenging. To simplify the behavioral analysis of machine learning models, Ribeiro et al. (2020) proposes *CheckList*, a list of inputs and expected outputs that aims to analyze general linguistic capabilities and NLP models mimicking the unit tests in software engineering. The integration of *CheckList* into UKP-SQuARE offers a simple method to analyze the performance of QA models beyond traditional benchmarks, such as MRQA tasks (Fisch et al., 2019).

As illustrated in Figure 4, we test the SQuAD 2.0 RoBERTa Adapter and SQuAD 2.0 BERT Adapter using the *CheckList* in which multiple NLP capabilities are tested like coreference, negation, and robustness. As we can see SQuAD 2.0 BERT Adapter performs worse than RoBERTa Adapter in the above dimensions. Such an example can be used by the lecturer in class to introduce the idea of behavioral tests on the fly. In addition, the behavioral tests of UKP-SQuARE can be used to foster the students' analytical skills. A potential assignment could be to train a QA model and deploy it on our platform to analyze it with the provided ecosystem of QA tools. In particular, thanks to the behavioral tests in UKP-SQuARE, students can provide a deeper analysis of their model based on the quantitative results of their test set and a quali-

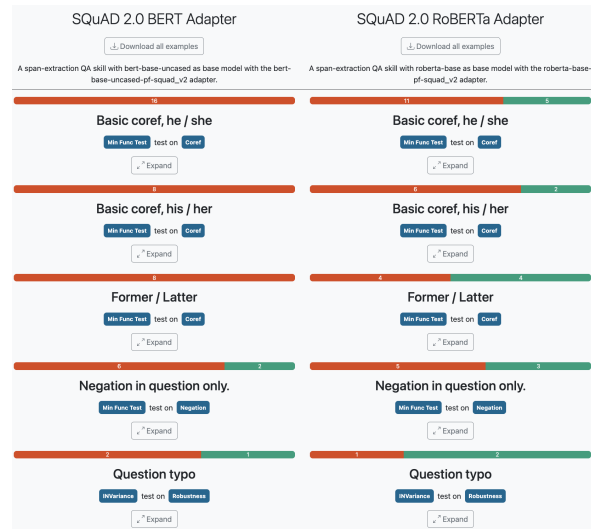


Figure 4: The result of running *CheckList* for SQuAD 2.0 RoBERTa Adapter and BERT Adapter. The number of failed and succeeded test cases are highlighted in green and red.

tative analysis based on the behavioral test results.

### 3.2.3 Adversarial Attacks

Policymakers are also designing a regulatory framework that guarantees users that their AI models are resilient to adversarial attacks<sup>5</sup>. Therefore, AI curriculums should also include adversarial attacks to prepare students for these new regulations.

UKP-SQuARE provides tools to conduct adversarial attacks, such as HotFlip (Ebrahimi et al., 2018), input reduction (Feng et al., 2018), and subspan (Jain et al., 2020). Thus, the home readings should include a theoretical introduction to these methods. Then, the lecture would use the platform to exploit the interactive nature of adversarial attacks. In particular, the need to analyze examples to understand different types of attacks makes this part of the topic especially practical. Therefore, the lecturer can introduce the topic through UKP-SQuARE and delve deeper into the technical details afterward.

An exemplary case is that students can attack real models with examples by tuning different parameters, such as the number of flips in HotFlip, to see how the output changes when they subtly change the input data. In Figure 5, only flipping *. (full stop)* to *wore* can directly change the answer. In class, a small experiment can be set up by lecturers in which students need to manually manipulate the input to see if it can trick the model into making

<sup>5</sup>See footnote 3

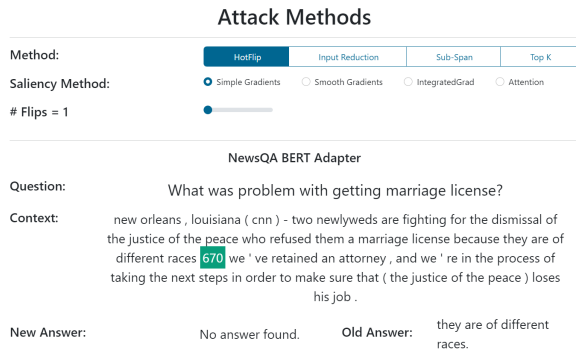


Figure 5: A HotFlip example where only flipping `.` (full stop) to `670` changes the answer.

incorrect answers and compare it with adversarial attack tools to deepen their understanding of those adversarial attacks and the importance of building up trustworthy QA systems.

### 3.2.4 Graph-based QA Models

Knowledge Graph Question Answering (KGQA) systems can have strong explanatory power thanks to the reasoning paths that can be extracted from the graph. Such transparency can enhance the interpretability and trustworthiness of the system. UKP-SQuARE currently offers QA-GNN (Yasunaga et al., 2021), a KGQA model that makes use of ConceptNet (Speer et al., 2017), and provides a visualization interface to explore the subgraph used by the model.

Although a reasoning path in a graph may provide a clear explanation of a model’s prediction, we believe that interpreting graph-based models is not straightforward because, usually, that path contains many irrelevant nodes and edges that may obscure the actual reasoning of the model. Thus, we propose to teach KGQA models with real examples of graphs. In this way, the lecturer, or even the students themselves, have to show the process of cleaning the graph to obtain and interpret the reasoning path. This process would be much more valuable for the future endeavor of the students than using a set of slides with examples of preprocessed clean graphs because they will be able to reproduce what they learn in real-use cases in companies.

## 3.3 Learning Multi-Agent Systems

Lastly, the current progress in QA is pushing toward creating robust models across multiple domains. To do this, there are two types of approaches: multi-dataset models and multi-agent models. While the former aims to train a single

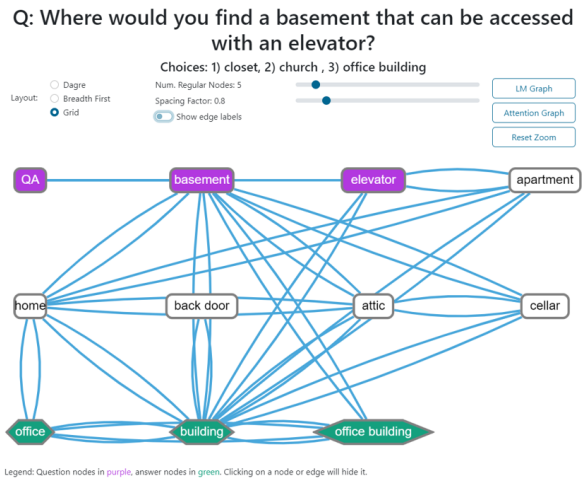


Figure 6: A visualized reasoning graph of the question *Where would you find a basement that can be accessed with an elevator?*

architecture on multiple datasets, the latter does the opposite. It trains multiple models (agents) on single datasets and combines the agents. UKP-SQuARE is compatible with both approaches; therefore, it is an ideal platform to teach them.

Thanks to UKP-SQuARE, we can also follow a flipped classroom methodology to teach multi-agent systems. After reading class materials explaining the models of this topic at home, the class time can be used as an explanation of the topic with a live demonstration of these models. In particular, we can easily show that multi-agent systems such as MetaQA (Puerto et al., 2021) select different agents depending on the input question. Figure 7 shows that the first answer selected by MetaQA, which is the correct one, is from an out-of-domain agent, while the second answer, which is not correct, is from the in-domain agent. This example illustrates the collaboration between agents achieved by multi-agent systems and can be an ideal way of starting the lecture on this topic before explaining the architectural details of MetaQA. Similarly, the platform can be used to introduce multi-dataset systems such as UnifiedQA (Khashabi et al., 2020), before delving into in-detail explanations of the model. In particular, the lecturer can explain the multiple accepted QA formats by UnifiedQA through real examples, and then, continue the explanation with the training details of the model with the support of slides.

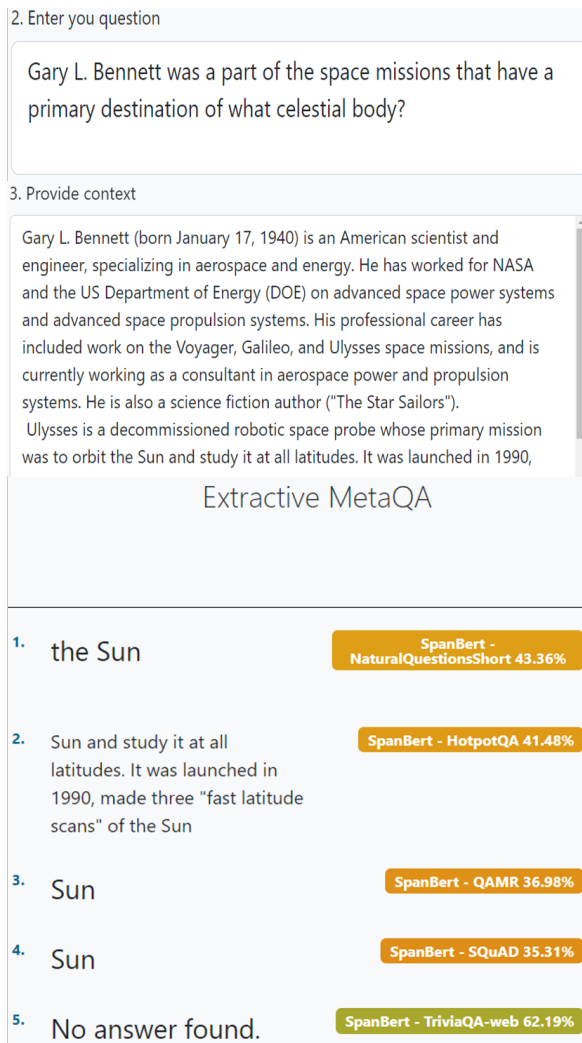


Figure 7: Multi-Agent QA in UKP-SQuARE: different agents are selected to predict the answer based on the input

### 3.4 Assignments with UKP-SQuARE

In addition to the above teaching scenarios in class, we also propose a homework assignment based on UKP-SQuARE<sup>6</sup> that leverages the insights and knowledge they acquire from the class. The students need to train their own QA model using the popular Hugging Face’s Transformer library (Wolf et al., 2020), deploy the model on our platform, and then write an in-detail report where they analyze their model from multiple perspectives. This report must include a quantitative analysis of the performance of their model on the test set and also a qualitative analysis that includes an explanation of the outputs of the model to a series of input questions, adversarial attacks that shows errors of their

<sup>6</sup>[https://colab.research.google.com/drive/17qw1dLWmU5EDxf9TLR29zIG9-EGKmNxP?usp=share\\_link](https://colab.research.google.com/drive/17qw1dLWmU5EDxf9TLR29zIG9-EGKmNxP?usp=share_link)

model, and an analysis of the possible behavioral errors obtain from *CheckList*. Furthermore, the students should also compare their model with other available models and identify the type of questions where their model fails. This would help them understand that models overfit the domain of their training data and, therefore, may fail in other domains. This assignment requires students to truly understand each component they learned during the class, which will help them consolidate their knowledge and develop a deeper understanding of the inner workings of different QA techniques. Additionally, the assignment can serve as a useful assessment tool, enabling teachers to gauge students’ understanding of the material and provide targeted feedback and support as needed.

### 3.5 User Study

To quantitatively evaluate the effectiveness of UKP-SQuARE in teaching the above QA techniques, we designed a questionnaire to collect feedback from students. The questionnaire was administered to a group of students who had completed a graduate NLP course that used our platform in both class time and for the assignment. All participants are 20-to-30 years-old graduate students in computer science. The questionnaire mainly focuses on two aspects: whether UKP-SQuARE deepens their understanding of techniques in QA systems and whether it makes it easier to get hands-on experience in UKP-SQuARE. The majority of questions require students to rate on a scale of 1 to 5. The complete questionnaire can be found in Appendix A.

Figure 8 shows the Likert scale chart with the responses of seven students who participated in the survey. As we can see, students have very positive attitudes towards all aspects of UKP-SQuARE for their QA learning. All participants think that the platform makes the class more engaging and interesting. In particular, most of them (91%) think UKP-SQuARE helps them better distinguish different QA formats. For information retrieval, the majority of the responders do not think that the platform can help them understand better the difference between lexical retrieval and semantic retrieval. The main reason behind this is that the difference between lexical and semantic retrievers is challenging to distinguish only via visualization unless students actively compare the documents by themselves. Besides, it also requires students



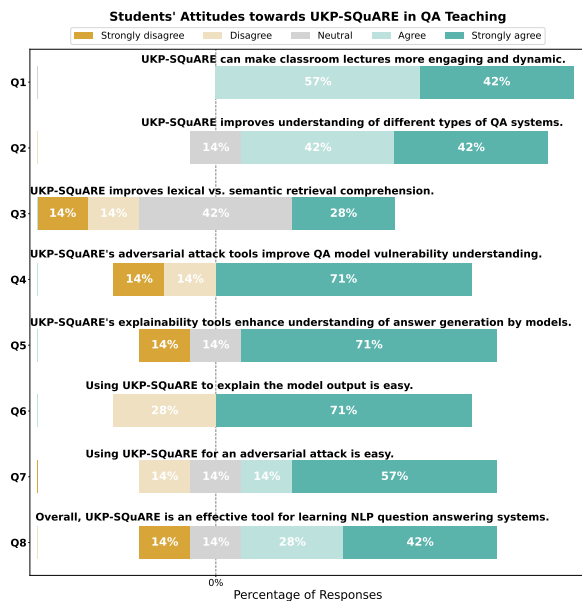


Figure 8: Students feedback towards UKP-SQuARE used in QA education.

to have a good understanding of semantic similarity and lexical similarity. Therefore, we plan to improve it by showing the difference between vector similarity and keyword matching between questions and retrieved documents. Regarding explainability and adversarial attack tools, around two-thirds of students believe that the platform facilitates their learning process of these topics. When it comes to hands-on experience, the vast majority of students agree that UKP-SQuARE is easy to use. Our platform provides an infrastructure that dramatically lowers the bar for students to get hands-on experience. All students think that without UKP-SQuARE, they would spend more time finding suitable open-source software to compare different models, analyze the output, and conduct adversarial attacks. Moreover, the respondents estimated that without UKP-SQuARE, the average time spent on homework would increase from 2-5 hours to more than 8 hours. One student also commented that doing experiments with the platform was straightforward and allowed him to try different ideas without any overhead. Therefore, although the survey sample is small and limits the conclusions, this overall positive feedback invites us to continue investigating how to conduct our QA and NLP classes more interactively with UKP-SQuARE and suggests that our students would benefit from extending this interactive class to other NLP topics such as generative pre-trained large language models, prompting with reinforcement

learning from human feedback, word embeddings, parsing trees, and machine translation among others.

## 4 Related Work

The most relevant tool is the AllenNLP demo<sup>7</sup>, which provides a user interface to the main components of the AllenNLP library (Gardner et al., 2018). This website includes an interface where users can interact with five extractive QA models. However, their goal is to have a showcase of their library rather than an extensive platform for teaching QA. Thus, their functionalities are limited. Most of their deployed models are outdated, only cover extractive QA settings, and do not provide information retrieval methods. Moreover, their explainability and adversarial attacks are not compatible with their transformer-based model. Furthermore, they do not provide graph-based models, which can be useful to explain graph neural networks and explainability methods based on graphs. Additionally, it cannot be used for our homework assignment because users cannot deploy and analyze their own models with explainability and adversarial attack tools as in our platform. However, they do provide demos for other NLP topics, such as Open Information Extraction and named entity recognition, and parsing trees, among others.

## 5 Conclusion

In this paper, we present a novel method to teach question-answering to postgraduate NLP students following the learner-centered method of flipped classrooms. We propose to provide reading materials to the students before the class and use the UKP-SQuARE platform as a driving tool to conduct the class. This platform integrates the most popular QA pipelines and an ecosystem of tools to analyze the available models. These tools include explainability methods, behavioral tests, adversarial attacks, and graph visualizations. We provide a series of use cases for teaching based on the provided models and methods by UKP-SQuARE, showing that classes can become much more interactive by using UKP-SQuARE than in conventional lectures. To evaluate the effectiveness of the platform and our methodology, we conducted a survey to collect feedback from students who took our class. The results show that most of the students think

<sup>7</sup><https://demo.allennlp.org/reading-comprehension/>

UKP-SQuARE accelerates their learning process and reduces the overhead to get hands-on experience. We plan to extend our platform to support prompting large language models, and therefore, we leave as future work creating a curriculum to teach prompting methods.

## Acknowledgements

We thank Max Eichler, Martin Tutek, Thomas Arnold, Tim Baumgärtner, and the anonymous reviewers for their insightful comments on a previous draft of this paper. This work has been funded by the German Research Foundation (DFG) as part of the UKP-SQuARE project (grant GU 798/29-1), the QASciInf project (GU 798/18-3), and by the German Federal Ministry of Education and Research and the Hessian Ministry of Higher Education, Research, Science and the Arts (HMWK) within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

## References

- Saleh Alhazbi. 2016. [Using flipped classroom approach to teach computer programming](#). In *2016 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, pages 441–444.
- Tim Baumgärtner, Kexin Wang, Rachneet Sachdeva, Gregor Geigle, Max Eichler, Clifton Poth, Hannah Sterz, Haritz Puerto, Leonardo F. R. Ribeiro, Jonas Pfeiffer, Nils Reimers, Gözde Şahin, and Iryna Gurevych. 2022. [UKP-SQUARE: An online platform for question answering research](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 9–22, Dublin, Ireland. Association for Computational Linguistics.
- Jacob Bishop and Matthew A Verleger. 2013. The flipped classroom: A survey of the research. In *2013 ASEE Annual Conference & Exposition*, pages 23–1200.
- Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. [HotFlip: White-box adversarial examples for text classification](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.
- Richard M Felder and Rebecca Brent. 2003. Designing and teaching courses to satisfy the abet engineering criteria. *Journal of Engineering education*, 92(1):7–25.
- Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. [Pathologies of neural models make interpretations difficult](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728, Brussels, Belgium. Association for Computational Linguistics.
- Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. [MRQA 2019 shared task: Evaluating generalization in reading comprehension](#). In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 1–13, Hong Kong, China. Association for Computational Linguistics.
- Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. [AllenNLP: A deep semantic natural language processing platform](#). In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.
- Sarthak Jain, Sarah Wiegrefe, Yuval Pinter, and Byron C. Wallace. 2020. [Learning to faithfully rationalize by construction](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4459–4473, Online. Association for Computational Linguistics.
- Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. [SpanBERT: Improving pre-training by representing and predicting spans](#). *Transactions of the Association for Computational Linguistics*, 8:64–77.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. [Dense passage retrieval for open-domain question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.
- Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hananeh Hajishirzi. 2020. [UNIFIEDQA: Crossing format boundaries with a single QA system](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.
- Haritz Puerto, Tim Baumgärtner, Rachneet Sachdeva, Haishuo Fang, Hao Zhang, Sewin Tariverdian, Kexin Wang, and Iryna Gurevych. 2023. [UKP-SQuARE v3: A Platform for Multi-Agent QA Research](#). *arXiv preprint arXiv:2303.18120*.
- Haritz Puerto, Gözde Gül Şahin, and Iryna Gurevych. 2021. [Metaqa: Combining expert agents for multi-skill question answering](#). *arXiv preprint arXiv:2112.01922*.
- 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. [Beyond accuracy: Behavioral testing of NLP models with CheckList](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.
- Stephen Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. 1995. [Okapi at trec-3](#). In *Overview of the Third Text REtrieval Conference (TREC-3)*, pages 109–126. Gaithersburg, MD: NIST.
- Rachneet Sachdeva, Haritz Puerto, Tim Baumgärtner, Sewin Tariverdian, Hao Zhang, Kexin Wang, Hosain Shaikh Saadi, Leonardo F. R. Ribeiro, and Iryna Gurevych. 2022. [UKP-SQuARE v2: Explainability and adversarial attacks for trustworthy QA](#). In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 28–38, Taipei, Taiwan. Association for Computational Linguistics.
- Sofia Serrano and Noah A. Smith. 2019. [Is attention interpretable?](#) In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.
- Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. [Deep inside convolutional networks: Visualising image classification models and saliency maps](#). In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*.
- Karen Sparck Jones. 1988. *A Statistical Interpretation of Term Specificity and Its Application in Retrieval*, page 132–142. Taylor Graham Publishing, GBR.
- Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI’17*, page 4444–4451. AAAI Press.
- Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. [Axiomatic attribution for deep networks](#). In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR.
- Jesse Vig. 2019. Bertviz: A tool for visualizing multi-head self-attention in the bert model. In *ICLR workshop: Debugging machine learning models*.
- 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.
- Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. [QA-GNN: Reasoning with language models and knowledge graphs for question answering](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, Online. Association for Computational Linguistics.

## A Questionnaire

The questionnaire includes two parts:

- Whether UKP-SQuARE deepens their understanding of QA topic. Some exemplary questions are:
  - Does UKP-SQuARE help you understand different types of QA systems better (e.g. extractive QA, abstractive QA)?
  - Does the adversarial attack tool in UKP-SQuARE help you understand the potential vulnerability of QA models better?
  - Does the explainability tool in UKP-SQuARE help you understand better how the model generates answers based on the input?
  - Does using UKP-SQuARE in the classroom make the lecture more dynamic and engaging?
- Whether UKP-SQuARE makes it easier to get hands-on experience. Some exemplary questions are:
  - How long did you spend on the assignment?
  - If you don’t use UKP-SQuARE, what will you use to finish your assignment (which involves comparing different models, and adversarial attacks)?
  - Without UKP-SQuARE, how long do you think you need to finish your assignment(including searching for platforms or building a small service by yourself)?

- How easy it is to use UKP-SQuARE to do adversarial attacks against models?
- How easy it is to use UKP-SQuARE to explain the model output?
- If you don't use UKP-SQuARE and you need to perform adversarial attacks on your model, would you be able to complete the assignment? If so, how much more difficult would it be?
- If you don't use UKP-SQuARE and you need to interpret the answers of your model using saliency maps, would you be able to do it? if so, how much more difficult would it be?
- Does UKP-SQuARE UI help you compare models easier? (eg: compared to using Jupyter Notebooks)?

# Exploring Effectiveness of GPT-3 in Grammatical Error Correction: A Study on Performance and Controllability in Prompt-Based Methods

Mengsay Loem, Masahiro Kaneko, Sho Takase, and Naoaki Okazaki

Tokyo Institute of Technology

{mengsay.loem, masahiro.kaneko}@nlp.c.titech.ac.jp  
sho.takase@linecorp.com, okazaki@c.titech.ac.jp

## Abstract

Large-scale pre-trained language models such as GPT-3 have shown remarkable performance across various natural language processing tasks. However, applying prompt-based methods with GPT-3 for Grammatical Error Correction (GEC) tasks and their controllability remains underexplored. Controllability in GEC is crucial for real-world applications, particularly in educational settings, where the ability to tailor feedback according to learner levels and specific error types can significantly enhance the learning process. This paper investigates the performance and controllability of prompt-based methods with GPT-3 for GEC tasks using zero-shot and few-shot setting. We explore the impact of task instructions and examples on GPT-3's output, focusing on controlling aspects such as minimal edits, fluency edits, and learner levels. Our findings demonstrate that GPT-3 could effectively perform GEC tasks, outperforming existing supervised and unsupervised approaches. We also showed that GPT-3 could achieve controllability when appropriate task instructions and examples are given.

## 1 Introduction

Grammatical Error Correction (GEC) is an essential application of Natural Language Processing (NLP) in educational settings, as it significantly enhances learners' language skills and writing performance (Kaneko et al., 2022). In real-world applications, controlling specific GEC settings, such as minimal and fluency edits and learner level-based corrections, is crucial to address diverse learning needs and scenarios (Napoles et al., 2017; Bryant et al., 2019; Flachs et al., 2020). Although recent GEC approaches based on supervised learning have achieved remarkable progress, they heavily rely on large training datasets comprising both genuine and pseudo data (Xie et al., 2018; Ge et al., 2018; Zhao et al., 2019; Lichtarge et al., 2019; Xu et al., 2019; Choe et al., 2019; Qiu et al., 2019; Grundkiewicz

et al., 2019; Kiyono et al., 2019; Grundkiewicz and Junczys-Dowmunt, 2019; Wang and Zheng, 2020; Zhou et al., 2020; Wan et al., 2020; Koyama et al., 2021a). Collecting such data for each specific setting is challenging and time-consuming, which limits the scalability of these methods in various learning situations.

Prompt-based methods utilize large-scale pre-trained language models (PLMs), such as GPT-3, and have demonstrated promising results in numerous NLP downstream tasks. These tasks include natural language inference, question answering, and summarization (Brown et al., 2020; Radford et al., 2019). Given the demand for control in GEC tasks across various settings, prompt-based methods are appealing because they deliver exceptional performance without needing extensive labeled data. Despite the success of prompt-based methods in multiple NLP tasks, their application to GEC remains under-explored. Although Coyne and Sakaguchi (2023) and Fang et al. (2023) have recently assessed prompt-based methods on select GEC benchmarks, a comprehensive analysis has yet to be conducted. This study aims to bridge this gap by concentrating on in-depth analyses of prompt-based methods and their controllability, aspects that have not been thoroughly investigated in previous research.

Our research seeks to address the following questions: 1) To what extent can PLMs using prompt-based methods solve GEC tasks? and 2) Is it possible to control GEC settings with prompts written in natural language using prompt-based methods?

In this work, we demonstrate that prompt-based methods with GPT-3 (Brown et al., 2020) achieve outstanding performance in GEC tasks (Section 3). In addition, the approach provides better control over the GEC process using task instructions and examples (Section 5). We conduct analyses to examine the impact of different types of task instructions on GPT-3's performance in both zero-shot

and few-shot setting, which emphasizing the importance of appropriate task instructions for GEC tasks (Section 4.1). Additionally, we investigate the effect of varying the number of examples in few-shot setting, and reveal that performance improves as the number of examples increases, albeit not strictly linearly (Section 4.2).

Furthermore, we explore the model’s controllability in various GEC scenarios, more specifically, its ability to concentrate on either minimal or fluency aspects (Section 5.1) and edits based on learner levels (Section 5.2). Experimental results indicate that task instructions alone may be sufficient to control editing without examples. However, we found that combining task instructions with examples resulted in more effective controlling performance. This indicates the importance of both task instruction and examples for better control of GEC settings using prompt-based methods, although the example set tends to have more importance.

## 2 Overall Experimental Settings

In this study, we designed a series of experiments using the prompt-based method with GPT-3 to evaluate the performance in GEC tasks. We utilized the GPT-3 model (`text-davinci-003`) through the API provided by OpenAI<sup>1</sup>. Our experiments were conducted in two settings: zero-shot and few-shot.

**Zero-shot** In the zero-shot setting, we assessed GPT-3’s ability to perform GEC tasks without any prior examples. We employed the following template for prompts in the zero-shot setting:

```
{task instruction}: {input text};  
output:_____
```

**Few-shot** For the few-shot setting, we implemented in-context learning as described by Brown et al. (2020). We provided the model with a few examples to guide its understanding of the GEC task. We randomly sampled pairs of examples from the training (or validation) sets of each experimental setting to serve as examples for the model. Details on the number and source of examples used in each experiment are described in the corresponding sections below. The template for prompts in the few-shot setting is as follows:

```
{task instruction}
```

<sup>1</sup><https://openai.com/blog/openai-api>

```
{example 1}  
...  
{example N}  
{input text}; output:_____
```

**Prompt** We used natural language text prompts for all our experiments. The task instruction within the prompt serves as a directive that informs the model about the desired outcome of each task. We varied the task instructions in both zero-shot and few-shot setting to examine the model’s adaptability to different phrasings (refer to Section 4.1). The instruction candidates employed in our prompt analyses are listed in Appendix A. Examples of task instructions include: `Correct the grammatical errors in the following sentence, Revise mistakes in this text, and Rewrite the following text with proper grammar.`

## 3 General Performance

To address research question 1) mentioned in Section 1, we investigated the overall performance of the prompt-based method with GPT-3 in GEC tasks. This investigation is particularly relevant given the increasing prevalence of GPT-3 in various NLP applications and the need to assess its potential capabilities for GEC tasks specifically.

### 3.1 Settings

We evaluated the performance of GPT-3 on three GEC test sets: JFLEG (Napoles et al., 2017), CoNLL2014 (Ng et al., 2014), and W&I+LOCNESS (Bryant et al., 2019; Granger, 1998) using both zero-shot and few-shot settings with 16 examples. We used examples from the training set of JFLEG, NUCLE (Dahlmeier et al., 2013), and W&I+LOCNESS as examples in the few-shot setting when evaluating with JFLEG, CoNLL2014, and W&I+LOCNESS test sets, respectively.

We compared our prompt-based methods to baselines, including supervised and unsupervised approaches. For the supervised approach, we trained a Transformer (big) using the settings described in Vaswani et al. (2017) and employed annotated data from multiple training sets. These sets included W&I+LOCNESS, FCE corpus (Yanakoudakis et al., 2011), Lang-8 Corpus of Learner English (Mizumoto et al., 2012), and NUCLE. After removing uncorrected sentence pairs, the train-

Method	JFLEG	CoNLL2014	W&I+LOCNESS
Transformer (big)	53.22	51.11	51.36
Grundkiewicz and Junczys-Dowmunt (2019)	56.18	44.23	47.89
Grundkiewicz et al. (2019)	–	26.76	–
ChatGPT zero-shot with CoT (Fang et al., 2023)	61.40	51.70	36.10
GPT-3 zero-shot	64.51	56.05	53.07
GPT-3 16-shot	<b>67.02</b>	<b>57.06</b>	<b>57.41</b>

Table 1: Comparison of GPT-3’s performance using both supervised and unsupervised approaches on the JFLEG, CoNLL2014, and W&I+LOCNESS test sets in zero-shot and few-shot settings, with 16 examples. The upper block of the table shows the results for the supervised approach, while the middle block shows the results for the unsupervised approaches. The scores are GLEU scores for JFLEG,  $F_{0.5}$  scores for CoNLL2014, and W&I+LOCNESS.

ing data used to train the Transformer model was approximately 600K pairs. For unsupervised approach, we compared our methods to previous work in the literature including Grundkiewicz and Junczys-Dowmunt (2019) and Grundkiewicz et al. (2019) where models were pre-trained with synthetic data. We also compared with the result of ChatGPT performance in zero-shot with chain-of-thought (CoT) reported in Fang et al. (2023).

### 3.2 Results

Table 1 shows the GLEU scores for JFLEG,  $F_{0.5}$  scores for CoNLL2014, and W&I+LOCNESS. From the table, GPT-3 performed competitively in the GEC tasks in both zero-shot and few-shot settings, outperforming the Transformer model in all test sets. In the zero-shot setting, GPT-3 surpassed the Transformer, with gains of about 11, 5, and 2 percentage points on JFLEG, CoNLL2014, and W&I+LOCNESS, respectively. The few-shot setting with 16 examples further improved GPT-3’s performance, indicating the model’s capability to adapt to the task with minimal examples quickly.

When comparing GPT-3 to unsupervised methods, we observe that GPT-3 outperforms other approaches in all test sets consistently. This comparison demonstrates the advantage of GPT-3 over existing unsupervised methods, even in the zero-shot setting. When comparing the performance of ChatGPT in the zero-shot setting with CoT, GPT-3 outperforms ChatGPT CoT in all three test sets. These results indicate GPT-3 is a more effective model for GEC tasks, especially in unsupervised settings.

## 4 Investigation on Prompt

In this section, we analyze the impact of different factors in prompt on the performance of GPT-3 in

GEC tasks. We focus on two factors: (1) the type of task instructions used and (2) the number of examples used in the few-shot settings. Our primary objective is to comprehend the influence of various factors in prompts to the models’ output, which will enable us to optimize GPT-3 more effectively for GEC tasks.

### 4.1 Effect of Task Instruction

In this section, we examine the effect of various types of task instructions on GPT-3’s performance in GEC tasks. We conduct evaluations using different task instructions in both zero-shot and few-shot settings.

#### 4.1.1 Settings

We created three types of task instructions, with ten candidates per type, following related work on natural language inference task (Webson and Pavlick, 2022). The types of task instructions are as follows (See Appendix A for details). We used the JFLEG validation set in this experiment.

**Instructive** instructions explicitly request the model to correct grammatical errors in the given text, such as `Correct grammatical errors in this sentence` and `Revise grammatical mistakes in the following text`.

**Misleading** instructions do not directly ask for grammar correction but instead require paraphrasing or rewriting, such as `Paraphrase the following sentence` and `Rewrite the following text to make it clearer`.

**Irrelevant** instructions are unrelated to grammar correction, such as `Translate the following sentence` and `Write a news headline about this sentence`.

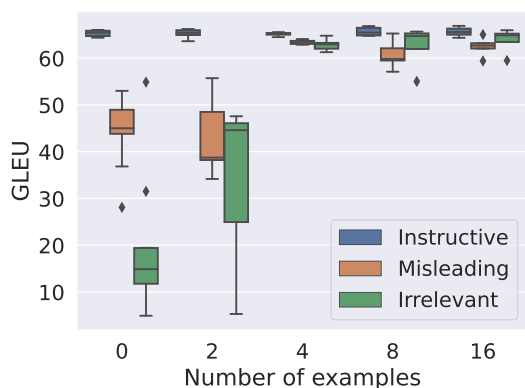


Figure 1: Comparison of GPT-3’s performance using different types of task instructions (Instructive, Misleading, and Irrelevant) in zero-shot and few-shot settings on GEC tasks.

#### 4.1.2 Result

Figure 1 shows the summary of the results when using different types of instructions in both zero-shot and few-shot settings. The findings reveal that task instructions significantly affect the performance of GPT-3 in GEC tasks.

In the zero-shot setting, instructive instructions produced the highest average score (65.54), while irrelevant instructions resulted in the lowest average score (17.05), clearly demonstrating that the type of task instruction impacts the model’s performance. Misleading instructions fell in the middle, with an average score of 43.45.

In few-shot settings, instructive instructions still outperformed the other two types, but the performance gap between instructive and misleading instructions decreased as the number of examples increased. The variance of the scores decreased with an increasing number of examples, suggesting that the model’s performance becomes more consistent as it receives more examples.

When comparing the different few-shot settings, we observed a clear trend of increasing performance as the number of examples increased. The standard deviation also decreased as the number of examples increased, indicating that the model’s performance became more consistent with more examples.

### 4.2 Effect of Number of Examples

In this section, we examine the impact of the number of examples used in few-shot settings on GPT-3’s performance. Our objective is to understand

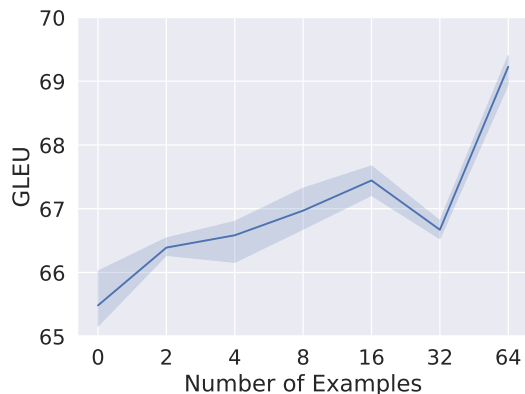


Figure 2: Effect of the number of examples on GPT-3’s performance in few-shot settings, evaluated on the JFLEG test set with a fixed task instruction.

how providing varying numbers of examples to the model influences its performance. By maintaining a fixed instruction and focusing solely on varying the number of examples, we aim to better comprehend their effect on the model’s performance.

#### 4.2.1 Settings

We conducted experiments on the JFLEG test set to examine the effect of the number of training examples on the model’s performance. The task instruction was kept consistent across all experiments. To perform the experiments, we randomly sampled examples from the training set of the JFLEG dataset. We tested the model with 2, 4, 8, 16, 32, and 64 examples, limiting the maximum number of examples to 64 due to the maximum input length of the model employed in our study.

#### 4.2.2 Result

The results obtained from each experimental setting are presented in Figure 2. Our experiments revealed a clear trend: performance improved as the number of examples increased. Our analysis further indicated that the models benefit from having more examples during the few-shot learning process. The highest score of 69.25 was achieved with 64 examples, suggesting that providing more examples can offer better guidance and context for the models to understand and effectively perform the task.

However, it is important to note that performance improvement is not strictly linear with the increase in the number of examples. For instance, the score slightly dipped from 67.11 to 66.67 when the number of examples increased from 16 to 32. This



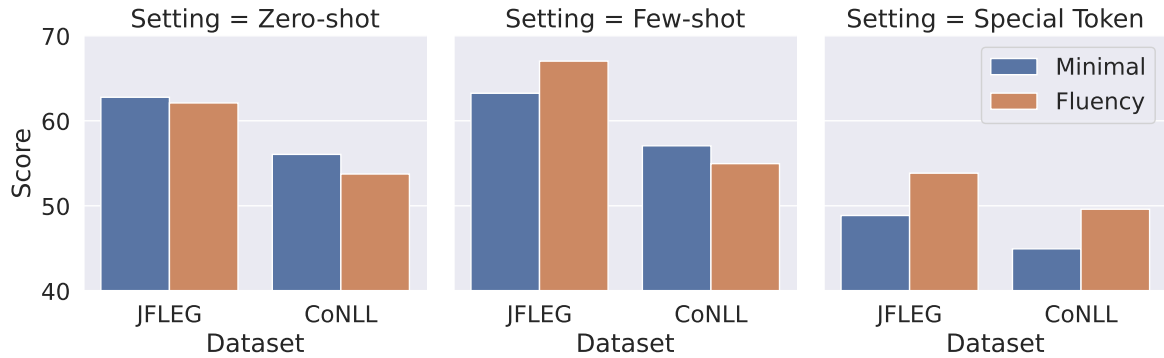


Figure 3: Comparison of GPT-3’s controllability for minimal and fluency edits using CoNLL2014 and JFLEG test sets, respectively, measured in GLEU scores.

deviation from linearity could be attributed to the quality of the examples or the inherent variability in the models’ performance. Further investigation is required to understand better the factors contributing to these fluctuations and identify the optimal number of examples needed to maximize performance.

## 5 Controllability through Prompt

In this section, we explore GPT-3’s controllability for GEC tasks through prompt-based methods. Our experiments focus on two settings: (1) comparing the model’s performance when instructed to make minimal edits versus emphasizing fluency, and (2) tailoring the editing to different learner levels, including beginner, intermediate, advanced, and native speakers. We aim to gain insights into GPT-3’s flexibility and controllability under various conditions. We also analyze the relative influence of task instruction and examples to identify the factor that significantly impacts the model’s output.

### 5.1 Minimal vs. Fluency Edits

#### 5.1.1 Settings

We evaluated controllability for minimal and fluency edits using the CoNLL2014 and JFLEG test sets, respectively. CoNLL2014 is a widely-used benchmark for GEC tasks, while JFLEG focuses on fluency-based evaluation. We conducted experiments in zero-shot and 16-shot settings. We used different task instructions to control the settings in the prompts, such as ‘Revise the following sentence with proper grammar’ for minimal edits and ‘Revise the following sentence to improve fluency’ for fluency edits.

We assessed the models using performance-based evaluation and edit distance-based evaluation. Performance-based evaluation measures the model’s error correction or fluency improvement ability, while edit distance-based evaluation quantifies the difference between original and revised sentences, offering insights into the extent of editing performed.

#### 5.1.2 Results

**Performance-based Evaluation** Figure 3 compares scores in performance-based evaluation for minimal and fluency edit instructions. In the zero-shot setting, minimal edit instructions perform better on the CoNLL2014 test set, while both instructions yield comparable scores on the JFLEG set. In the few-shot setting, higher scores are observed when using corresponding task instructions for each test set, emphasizing the effectiveness of text prompts in controlling editing settings. The discrepancy between zero-shot and few-shot settings might be due to the model’s limited understanding of the task in the zero-shot setting. Additional examples in the few-shot setting enable the model to comprehend the task’s objective better and adjust its output accordingly.

Additionally, we also compared the prompt-based method with a supervised controlling method that uses special tokens as in Johnson et al. (2017), where different special tokens were used to control target languages in multilingual translation. We trained a Transformer (Big) encoder-decoder with annotated data tagged with special tokens indicating minimal and fluency edits settings. Despite using more training data, this supervised method failed to control specific settings while achieving higher scores on both test sets with fluency edit

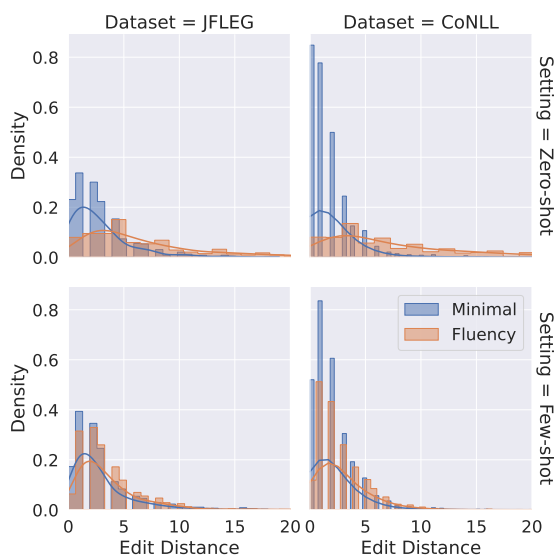


Figure 4: Edit distance distributions for minimal and fluency edits on CoNLL2014 and JFLEG test sets, respectively, as part of the edit distance-based evaluation for controllability of prompts.

tokens, as in Figure 3. This finding highlights the potential advantages of the prompt-based approach.

**Edit Distance-based Evaluation** Figure 4 presents edit distance distributions for each setting as part of edit distance-based evaluation. A shift to the right indicates more edits performed with fluency edit instructions. In the few-shot setting, the difference in edit distance distributions between minimal and fluency edits is smaller than in the zero-shot setting, which can be attributed to the influence of the examples presented in the prompt. The model’s ability to generalize from examples in the few-shot setting may diminish the difference in edit distance between the two settings, further emphasizing the importance of carefully selected examples.

In summary, the prompt-based method using GPT-3 can effectively control GEC task outputs for either minimal or fluency edits. Controllability is more evident in few-shot settings, where additional examples help the model adapt its behavior according to the given instructions. The edit distance-based evaluation further supports the model’s ability to adjust its editing behavior based on the prompt, showcasing its potential for practical applications.

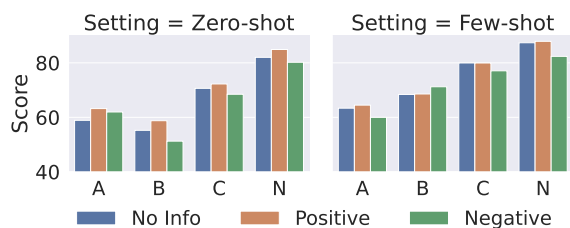


Figure 5: Impact of task instructions with varying additional information on GPT-3’s performance in GEC tasks, evaluated on the validation sets of W&I+LOCNESS. The experiment features three settings: No Info, Positive Info, and Negative Info. The x-axis represents different CEFR levels (A, B, C) and native speakers (N) included in the validation set.

## 5.2 Learner Level-based Correction

### 5.2.1 Settings

In this section, we examine GPT-3’s adaptability to diverse GEC task requirements and contexts by analyzing the impact of varying additional information in task instructions. We conducted experiments in both zero-shot and few-shot (16-shot) settings. We utilized the W&I+LOCNESS validation sets, comprising text from various CEFR levels (A: Beginner, B: Intermediate, C: Advanced) and native speakers (N) as evaluation sets. We devised an experiment with three settings based on the following types of additional information (refer to Appendix B):

**No Info:** No extra information is provided.

**Positive Info:** Information that supports the input sentence’s characteristics, such as the number of errors to be revised. Example: "Revise mistakes in the following text written by a beginner learner with a lot of mistakes."

**Negative Info:** Information that contrasts with the input sentence’s characteristics, e.g., a text written by a beginner learner with many errors but described as having few. Example: "Revise mistakes in the following text written by an advanced learner with only a few mistakes."

### 5.2.2 Results

Figure 5 shows the results of controlling task instruction with additional information on learner levels. In the zero-shot setting, positive information improved performance, while negative information adversely impacted output across most learner levels. This demonstrates the influence of additional

information in task instructions. In the few-shot setting, task instructions without additional information (No Info) achieved comparable scores to cases with Positive Info, suggesting that the model effectively utilizes examples to understand the desired correction level. However, with Negative Info, performance dropped for most learner levels compared to No Info and Positive Info cases.

### 5.3 Effect of Task Instruction vs. Examples

In this section, we present an experiment to examine the relative effect of task instruction and examples on GPT-3’s performance in controllability, in few-shot settings. Our primary objective is to determine which of these two components, task instruction and example, has a more significant impact on the model’s outputs. Moreover, we extend our investigation to explore the influence of examples on the editing process of the output, providing a more comprehensive understanding of the interplay between these variables in the context of few-shot learning.

#### 5.3.1 Settings

To investigate the relative influence of task instructions and examples independently, we designed two experiments, each featuring distinct conditions:

**Varied Task Instruction with Fixed Examples (VIFE)** We modified the task instructions while maintaining a constant set of examples. This approach allows us to assess the influence of task instructions on the model’s performance.

**Fixed Task Instruction with Varied Examples (FIVE)** We utilized a single task instruction and altered the set of examples. This condition helps us evaluate the impact of examples on the model’s performance.

In this experiment, we employed the JFLEG and CoNLL2014 test sets. We assessed the performance using  $F_{0.5}$  score for CoNLL2014 and GLEU for JFLEG. For the VIFE condition, we prepared a fixed set of examples and a varied set of task instructions for each dataset, similar to the approach in Section 5.1. We used task instructions that requested the model to perform minimal edits on the CoNLL2014 test set and fluency edits on the JFLEG test set. For the FIVE condition, we prepared fixed task instructions and varied examples from the training sets of NUCLE and JFLEG, which correspond to minimal and fluency edits,

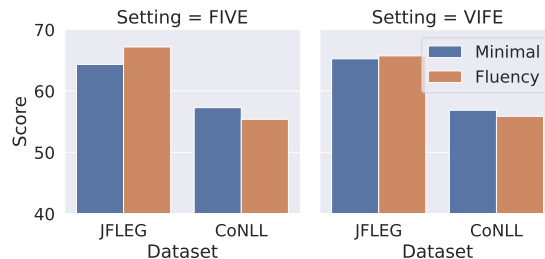


Figure 6: Comparison of the impact of task instructions and number of examples in few-shot settings. VIFE condition examines the effect of varied task instructions with fixed examples, while FIVE condition evaluates the impact of fixed task instructions with varied examples.

Test set	Example from	
	JFLEG	NUCLE
	Fluency Edits	
JFLEG	<b>0.1569</b>	0.1893
CoNLL2014	0.4443	<b>0.4058</b>
	Minimal Edits	
JFLEG	<b>0.2283</b>	0.3038
CoNLL2014	0.4158	<b>0.3768</b>

Table 2: Impact of example set on GPT-3’s performance in few-shot settings evaluated on JFLEG and CoNLL2014 test sets, measured by Jensen-Shannon distance. Diagonal entries show closer alignment between model output and corresponding example set.

respectively. We conducted experiments in this section with 16-shot setting.

#### 5.3.2 Results

Figure 6 summarizes the results regarding the performance scores. In both CoNLL2014 and JFLEG, we observed performance gaps between the two settings, minimal and fluency edits. However, the gaps were more drastic when changing the example set compared to varying the task instruction. These results suggest that examples play a more critical role in controlling the model’s behavior than task instructions, as changing the example set leads to more significant differences in achieving the desired output. This is likely because examples provide specific and contextual information, while task instructions can be abstract and open to interpretation. This highlights the importance of carefully selecting examples to optimize model performance.

We further investigated the example set’s impact on model output, using Jensen-Shannon distance to compare edit distance distributions in both minimal

and fluency edits settings. Lower Jensen-Shannon distance indicates a more similar edit distribution between the example set and model output. Results in Table 2 show lower distances in diagonal entries, signifying closer alignment between the model output and corresponding example set. This highlights the importance of carefully selecting examples to guide the model in generating outputs with desired characteristics.

## 6 Related Work

Supervised learning approaches have predominantly driven GEC research, resulting in state-of-the-art performance. Encoder-decoder models are commonly employed in GEC using supervised learning. Yuan and Briscoe (2016) first applied an encoder-decoder model to GEC, inspiring subsequent researchers to propose various encoder-decoder-based GEC models (Ji et al., 2017; Chollampatt and Ng, 2018; Junczys-Dowmunt et al., 2018; Zhao et al., 2019; Kaneko et al., 2020; Yamashita et al., 2020). These methods typically rely on large training datasets containing parallel sentences with and without grammatical errors (Kiyono et al., 2019). However, scalability remains challenging, as labeled data is required for each specific situation, such as grammar correction style or input text domain.

Unsupervised GEC approaches aim to reduce dependency on labeled data by leveraging unsupervised learning techniques, including PLMs, hand-crafted rules, denoising autoencoders, or unsupervised machine translation (Grundkiewicz et al., 2019; Grundkiewicz and Junczys-Dowmunt, 2019; Flachs et al., 2019; Solyman et al., 2021; Koyama et al., 2021b). However, these methods necessitate creating large-scale pseudo data for model training, making it difficult to generate pseudo-data and train models for different learning scenarios. Some studies have proposed unsupervised GEC methods using PLMs (Alikaniotis and Raheja, 2019; Yasunaga et al., 2021), but they have not focused on prompt-based methods with PLMs.

Recently, the GPT-3 model (Brown et al., 2020) has demonstrated remarkable performance across various NLP tasks, although its GEC performance remains limited. Schick et al. (2022) employed a simple zero-shot prompt for GEC, while Dwivedi-Yu et al. (2022) conducted a more comprehensive analysis using diverse zero-shot prompts. Coyne and Sakaguchi (2023) and Fang et al. (2023) com-

pared the latest GPT-3 model’s performance (text-davinci-003) and ChatGPT against GEC leaderboard models and reference edits, finding that these prompt-based methods exhibited strong GEC performance. However, automatic metrics and human evaluations occasionally disagreed on the relative quality of corrections.

Controlling GEC model generation is crucial but remains underexplored. Hotate et al. (2019) proposed a GEC method that controls the degree of correction by tagging input with the correction level, but it requires supervised learning with parallel data. Additionally, Hotate et al. (2020) suggested a beam search method to control GEC correction diversity by dynamically updating search tokens within the beam based on the likelihood of predicting source sentence tokens. While this method enables model control without additional training, it falls short in accommodating specific learner requests, such as minimal and fluency edits.

GEC model evaluation methods have been proposed based on learner levels and correction styles. To account for differences in correction styles and domains, Maeda et al. (2022) introduced a method to train evaluation models using only parallel data. Takahashi et al. (2022) created proficiency-annotated data to train evaluation models and developed an evaluation method that considers proficiency by fine-tuning PLMs (Yoshimura et al., 2020).

## 7 Conclusion

In conclusion, this study demonstrates the potential of using prompt-based methods with GPT-3 for GEC tasks, achieving competitive performance compared to traditional supervised and unsupervised methods. By carefully crafting task instructions and examples, we show that GPT-3 can be effectively controlled to focus on different aspects of the GEC process and adapt to diverse learning needs. Our findings highlight the importance of optimizing task instructions and example selection to enhance the performance and controllability of GPT-3, paving the way for further research on refining prompt engineering techniques and exploring their applicability to other NLP tasks and language models.

## 8 Educational Implications and Community Benefits

Our study provides valuable implications for education. The controllability of large-scale language models in GEC tasks can be leveraged to design personalized language instruction. It allows educators to provide feedback that matches individual students' proficiency levels and focuses on specific areas for improvement. For learners, instant, tailored feedback can enhance their language learning process. Moreover, our findings can improve intelligent tutoring systems, making them more responsive to individual needs. Beyond education, our research can enhance language-based interfaces and AI communication systems, offering more accurate and context-specific language corrections. This study lays the groundwork for future exploration into how large language models can improve language education and literacy.

## 9 Limitation

While our study provides valuable insights into the use of prompt-based methods with GPT-3 for GEC tasks and its controllability, several limitations should be acknowledged.

**Focus on GPT-3:** This study exclusively examines GPT-3 as the language model for GEC tasks. While GPT-3 has shown remarkable performance in various NLP tasks, other pre-trained language models, such as GPT-4, may offer different results. A broader investigation that includes other language models would provide a more comprehensive understanding of the applicability of prompt-based methods in GEC tasks.

**Limited evaluation metrics:** The evaluation of GPT-3's performance and controllability in our experiments mainly relies on quantitative metrics, such as edit distance and task scores. These metrics may not fully capture the nuances of grammatical error correction or the model's ability to adapt to different learning scenarios. Additional qualitative analysis, along with more diverse evaluation metrics, could provide a richer understanding of the model's performance and controllability.

**Variability in examples:** While our study highlights the importance of example selection in few-shot settings, we do not thoroughly explore the impact of example quality or diversity. The effect

of using different types of examples or a more diverse set of examples remains to be investigated, which could further inform the design of effective example sets for prompt-based GEC tasks. By addressing these limitations in future research, we can further advance our understanding of the performance and controllability of prompt-based methods with GPT-3 and other language models in GEC tasks and beyond.

**Potential fine-tuning on test data:** There is a possibility that GPT-3 has been fine-tuned (instruction tuning) on the test data we are using, which might explain the higher evaluation scores compared to previous research. As this information has not been disclosed, we are unable to verify it at this time. This point should be taken into consideration when interpreting our results.

## Acknowledgements

These research results were obtained partially from the commissioned research (No. 225) by National Institute of Information and Communications Technology (NICT), Japan.

## References

- Dimitris Alikaniotis and Vipul Raheja. 2019. [The unreasonable effectiveness of transformer language models in grammatical error correction](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 127–133, Florence, Italy. Association for Computational Linguistics.
- 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.
- Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. [The BEA-2019 shared task on grammatical error correction](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.
- Yo Joong Choe, Jiyeon Ham, Kyubyong Park, and Yeol Yoon. 2019. [A neural grammatical error correction](#)

- system built on better pre-training and sequential transfer learning. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 213–227, Florence, Italy. Association for Computational Linguistics.
- Shamil Chollampatt and Hwee Tou Ng. 2018. A Multilayer Convolutional Encoder-Decoder Neural Network for Grammatical Error Correction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5755–5762, New Orleans, Louisiana. Association for the Advancement of Artificial Intelligence.
- Steven Coyne and Keisuke Sakaguchi. 2023. An analysis of gpt-3’s performance in grammatical error correction.
- Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner English: The NUS corpus of learner English. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31, Atlanta, Georgia. Association for Computational Linguistics.
- Jane Dwivedi-Yu, Timo Schick, Zhengbao Jiang, Maria Lomeli, Patrick Lewis, Gautier Izacard, Edouard Grave, Sebastian Riedel, and Fabio Petroni. 2022. Editeval: An instruction-based benchmark for text improvements. *ArXiv*, abs/2209.13331.
- Tao Fang, Shu Yang, Kaixin Lan, Derek F. Wong, Jinpeng Hu, Lidia S. Chao, and Yue Zhang. 2023. Is chatgpt a highly fluent grammatical error correction system? a comprehensive evaluation.
- Simon Flachs, Ophélie Lacroix, and Anders Søgaard. 2019. Noisy channel for low resource grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 191–196, Florence, Italy. Association for Computational Linguistics.
- Simon Flachs, Ophélie Lacroix, Helen Yannakoudakis, Marek Rei, and Anders Søgaard. 2020. Grammatical error correction in low error density domains: A new benchmark and analyses. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8467–8478, Online. Association for Computational Linguistics.
- Tao Ge, Furu Wei, and Ming Zhou. 2018. Fluency boost learning and inference for neural grammatical error correction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1055–1065, Melbourne, Australia. Association for Computational Linguistics.
- Sylviane Granger. 1998. The computer learner corpus: a versatile new source of data for sla research.
- Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2019. Minimally-augmented grammatical error correction. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 357–363, Hong Kong, China. Association for Computational Linguistics.
- Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263, Florence, Italy. Association for Computational Linguistics.
- Kengo Hotate, Masahiro Kaneko, Satoru Katsumata, and Mamoru Komachi. 2019. Controlling grammatical error correction using word edit rate. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 149–154, Florence, Italy. Association for Computational Linguistics.
- Kengo Hotate, Masahiro Kaneko, and Mamoru Komachi. 2020. Generating diverse corrections with local beam search for grammatical error correction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2132–2137, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Jianshu Ji, Qinlong Wang, Kristina Toutanova, Yongen Gong, Steven Truong, and Jianfeng Gao. 2017. A nested attention neural hybrid model for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 753–762, Vancouver, Canada. Association for Computational Linguistics.
- Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.
- Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. Approaching neural grammatical error correction as a low-resource machine translation task. 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 595–606, New Orleans, Louisiana. Association for Computational Linguistics.
- Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020. Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4248–4254, Online. Association for Computational Linguistics.
- Masahiro Kaneko, Sho Takase, Ayana Niwa, and Naoaki Okazaki. 2022. Interpretability for language learners

- using example-based grammatical error correction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7176–7187, Dublin, Ireland. Association for Computational Linguistics.
- Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. [An empirical study of incorporating pseudo data into grammatical error correction](#). 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 1236–1242, Hong Kong, China. Association for Computational Linguistics.
- Aomi Koyama, Kengo Hotate, Masahiro Kaneko, and Mamoru Komachi. 2021a. [Comparison of grammatical error correction using back-translation models](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 126–135, Online. Association for Computational Linguistics.
- Shota Koyama, Hiroya Takamura, and Naoaki Okazaki. 2021b. [Various errors improve neural grammatical error correction](#). In *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation*, pages 251–261, Shanghai, China. Association for Computational Linguistics.
- Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. [Corpora generation for grammatical error correction](#). 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 3291–3301, Minneapolis, Minnesota. Association for Computational Linguistics.
- Koki Maeda, Masahiro Kaneko, and Naoaki Okazaki. 2022. [IMPARA: Impact-based metric for GEC using parallel data](#). In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3578–3588, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Tomoya Mizumoto, Yuta Hayashibe, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2012. [The effect of learner corpus size in grammatical error correction of ESL writings](#). In *Proceedings of COLING 2012: Posters*, pages 863–872, Mumbai, India. The COLING 2012 Organizing Committee.
- Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. [JFLEG: A fluency corpus and benchmark for grammatical error correction](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 229–234, Valencia, Spain. Association for Computational Linguistics.
- Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. [The CoNLL-2014 shared task on grammatical error correction](#). In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.
- Mengyang Qiu, Xuejiao Chen, Maggie Liu, Krishna Parvathala, Apurva Patil, and Jungyeul Park. 2019. [Improving precision of grammatical error correction with a cheat sheet](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 240–245, Florence, Italy. Association for Computational Linguistics.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Timo Schick, Jane Dwivedi-Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave, and Sebastian Riedel. 2022. [Peer: A collaborative language model](#).
- Aiman Solyman, Wang Zhenyu, Tao Qian, Arafat Abdulgader Mohammed Elhag, Muhammad Toseef, and Zeinab Aleibeid. 2021. Synthetic data with neural machine translation for automatic correction in arabic grammar. *Egyptian Informatics Journal*, 22(3):303–315.
- Yujin Takahashi, Masahiro Kaneko, Masato Mita, and Mamoru Komachi. 2022. [ProQE: Proficiency-wise quality estimation dataset for grammatical error correction](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5994–6000, Marseille, France. European Language Resources Association.
- 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.
- Zhaohong Wan, Xiaojun Wan, and Wenguang Wang. 2020. [Improving grammatical error correction with data augmentation by editing latent representation](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2202–2212, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Lihao Wang and Xiaoqing Zheng. 2020. [Improving grammatical error correction models with purpose-built adversarial examples](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2858–2869, Online. Association for Computational Linguistics.
- Albert Webson and Ellie Pavlick. 2022. [Do prompt-based models really understand the meaning of their](#)

- prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2300–2344, Seattle, United States. Association for Computational Linguistics.
- Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Ng, and Dan Jurafsky. 2018. [Noising and denoising natural language: Diverse backtranslation for grammar correction](#). 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 619–628, New Orleans, Louisiana. Association for Computational Linguistics.
- Shuyao Xu, Jiehao Zhang, Jin Chen, and Long Qin. 2019. [Erroneous data generation for grammatical error correction](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 149–158, Florence, Italy. Association for Computational Linguistics.
- Ikumi Yamashita, Satoru Katsumata, Masahiro Kaneko, Aizhan Imankulova, and Mamoru Komachi. 2020. [Cross-lingual transfer learning for grammatical error correction](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4704–4715, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. [A new dataset and method for automatically grading ESOL texts](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.
- Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2021. [LM-critic: Language models for unsupervised grammatical error correction](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7752–7763, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Ryoma Yoshimura, Masahiro Kaneko, Tomoyuki Kajiwara, and Mamoru Komachi. 2020. [SOME: Reference-less sub-metrics optimized for manual evaluations of grammatical error correction](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6516–6522, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Zheng Yuan and Ted Briscoe. 2016. [Grammatical error correction using neural machine translation](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386, San Diego, California. Association for Computational Linguistics.
- Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. [Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data](#). 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 156–165, Minneapolis, Minnesota. Association for Computational Linguistics.
- Wangchunshu Zhou, Tao Ge, Chang Mu, Ke Xu, Furu Wei, and Ming Zhou. 2020. [Improving grammatical error correction with machine translation pairs](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 318–328, Online. Association for Computational Linguistics.



## **A Prompts for Investigation on Instruction Effect**

All instructions used for experiments described in Section 4.1 are listed in Table 3.

## **B Prompts for Learner's Level-based Control**

All instructions and additional information used for experiments described in Section 5.2 are listed in Table 4.

Type	Task Instruction
Instructive	<p>Correct grammatical errors in this sentence</p> <p>Revise grammatical mistakes in the following text.</p> <p>Edit this paragraph for grammar mistakes.</p> <p>Find and fix any errors in this sentence.</p> <p>Rewrite this sentence to correct its grammar.</p> <p>Identify and correct the grammar errors in this text.</p> <p>Make any necessary grammar corrections to this passage.</p> <p>Correct the grammar in this sentence without changing its meaning.</p> <p>Find and correct the errors in this paragraph.</p> <p>Proofread this text and correct any grammar mistakes.</p>
Misleading	<p>Paraphrase the following sentence.</p> <p>Rewrite the following text to make it clearer.</p> <p>Revise this paragraph to improve its clarity.</p> <p>Clarify the meaning of this sentence by rephrasing it.</p> <p>Make this sentence more concise without changing its meaning.</p> <p>Improve the readability of this text by rewording it.</p> <p>Reconstruct this sentence to enhance its clarity.</p> <p>Paraphrase this text to make it more comprehensible.</p> <p>Rewrite this paragraph to convey the same information in a clearer way.</p> <p>Edit this sentence to improve its coherence and flow.</p>
Irrelevant	<p>Translate the following sentence in to Japanese.</p> <p>Write a news headline about this sentence.</p> <p>Create a meme based on the following text.</p> <p>Write a short story based on this sentence.</p> <p>Compose a poem using the words in this paragraph.</p> <p>Write a summary of this text.</p> <p>Analyze the use of metaphor in this sentence.</p> <p>Explain the historical context of this passage.</p> <p>Write a tweet about this text.</p> <p>Write a letter to your future self based on the following sentence.</p>

Table 3: Prompts for Instruction Effect Investigation, showing three types of task instructions with ten candidate prompts each. The types include Instructive, Misleading, and Irrelevant prompts.

Info	Task Instruction
<b>Beginner</b>	
No Info	Revise mistakes in the following text
Positive Info	Revise mistakes in the following text written by a beginner learner with a lot of mistakes
Negative Info	Revise mistakes in the following text written by an advanced learner with only a few mistakes
<b>Intermediate</b>	
No Info	Revise mistakes in the following text
Positive Info	Revise mistakes in the following text written by an intermediate learner with some mistakes
Negative Info	Revise mistakes in the following text written by a native speaker
<b>Advanced</b>	
No Info	Revise mistakes in the following text
Positive Info	Revise mistakes in the following text written by an advanced learner with only a few mistakes
Negative Info	Revise mistakes in the following text written by a beginner learner with a lot of mistakes
<b>Native</b>	
No Info	Revise mistakes in the following text
Positive Info	Revise mistakes in the following text written by a native speaker
Negative Info	Revise mistakes in the following text written by a beginner learner with a lot of mistakes

Table 4: All prompts used in experiments investigating the controllability of learner level-based edits.

# A Closer Look at $k$ -Nearest Neighbors Grammatical Error Correction

Justin Vasselli and Taro Watanabe

Nara Institute of Science and Technology

{vasselli.justin\_ray.vk4, taro}@is.naist.jp

## Abstract

In various natural language processing tasks, such as named entity recognition and machine translation, example-based approaches have been used to improve performance by leveraging existing knowledge. However, the effectiveness of this approach for Grammatical Error Correction (GEC) is unclear. In this work, we explore how an example-based approach affects the accuracy and interpretability of the output of GEC systems and the trade-offs involved. The approach we investigate has shown great promise in machine translation by using the  $k$  nearest translation examples to improve the results of a pretrained Transformer model. We find that using this technique increases precision by reducing the number of false positives, but recall suffers as the model becomes more conservative overall. Increasing the number of example sentences in the datastore does lead to better performing systems, but with diminishing returns and a high decoding cost. Synthetic data can be used as examples, but the effectiveness varies depending on the base model. Finally, we find that finetuning on a set of data may be more effective than using that data during decoding as examples.

## 1 Introduction

Grammatical Error Correction (GEC) is the task of identifying and correcting grammatical mistakes in ungrammatical text. While it can be used to assist native speakers as well, it is frequently applied to text written by language learners, and can be used pedagogically to help them improve their writing skills. Providing feedback on grammatical errors in a learner’s writing allows them to learn from their mistakes and improve their writing over time. For this feedback to be effective, it must be interpretable to the learner.

GEC models are often based on neural machine translation (NMT) models and treated as similar to sequence-to-sequence (seq2seq) tasks (Junczys-

Dowmunt et al., 2018; Kiyono et al., 2019). Unfortunately, Transformer-based seq2seq models produce corrections that are uninterpretable, because they simply output a corrected sentence without any indication of how or why elements of the sentence were corrected. This lack of interpretability can make it difficult for learners to understand the nature of their mistakes and how to avoid them in the future. In contrast, example-based approaches to GEC can provide a motivating example for each correction, making the results more interpretable and therefore more helpful for learners. This can make the difference between a learner simply correcting a mistake and actually understanding why it is a mistake and how to avoid it in the future.

Example-based, or instance-based, methods have recently been applied to tasks across the field such as named entity recognition (Ouchi et al., 2020), summarization (Cao et al., 2018), and machine translation (Khandelwal et al., 2020). In their recent work, Kaneko et al. (2022) presented their findings on the interpretability of GEC corrections using human evaluation and three example selection methods: token-based retrieval, BERT-based retrieval, and their example-based grammatical error correction (EB-GEC) system. The study found that presenting examples is more useful to learners than providing none, with EB-GEC providing the most useful examples for language learners’ understanding and acceptance of the model corrections.

EB-GEC is based on the  $k$ -nearest neighbors approach to machine translation proposed by Khandelwal et al. (2020). This method uses a datastore constructed from a set of example sentence pairs during the decoding of the vanilla Transformer. At each timestep, the vector being passed into the final feedforward network of the decoder is used to locate the  $k$  nearest neighbor examples in the datastore. This vector represents the translation context, which is composed of the ungrammatical sentence plus the prefix of the output. The datas-

tore itself is constructed from a corpus, authentic or synthetic, of training examples. One entry into the datastore is made for each token of the corrected sentence of each example pair, using the encoded translation context as the key, and the ground-truth token as the value. During inference, the retrieved values of the  $k$  nearest contexts form a distribution of target tokens. The distribution of target tokens collected from the datastore is then interpolated with the distribution from the base Transformer. In this way, the output of the vanilla Transformer is influenced by the most similar examples from the datastore, and motivating examples are returned for each token of the output.

Khandelwal et al. (2020) reports high BLEU score gains in resource-rich languages with large databases, but less impressive performance in low-resource languages. Treating GEC as a low-resource machine translation task was proposed by Brockett et al. (2006) and has resulted in many high performing systems (Junczys-Dowmunt et al., 2018). However, there are key differences between grammatical error correction and machine translation. GEC is a monolingual task, where both input and output share a vocabulary, and a large number of tokens from the input sentence remain unchanged in the output sentence. This difference may very well affect the viability of using  $k$ -nearest neighbors for grammatical error correction.

Kaneko et al. (2022) found that their EB-GEC system improved the  $F_{0.5}$  score on three out of four test sets, relative to the vanilla Transformer. However, there are several factors to consider about these results. The three test sets that performed better using EB-GEC came from datasets with training splits used for both training the vanilla Transformer and as the datastore of example sentences. The fourth test set that performed better with the vanilla Transformer did not have any representation in the datastore or the training. This may indicate that EB-GEC is not generalizable, as in a way, the three test sets with better scores can be thought of as in-domain, because they had similar sentences used in the datastore. It is possible that using example sentence pairs produced in a different context would produce lower scores on the test sets. It is worth investigating how using different data for the example corrections than during training affects the results.

The reported scores of this system are lower than those reported by Kiyono et al. (2019), using a

vanilla Transformer pretrained on synthetic data. It is unclear whether applying the same kNN method to a higher performing Transformer would yield the same gains. A detailed analysis of the costs and benefits of using the  $k$ -nearest neighbor approach to grammatical error correction as proposed by Kaneko et al. (2022) has yet to be carried out, but the results of the initial experiments are worth investigating further.

This work aims to address some outstanding questions about the effectiveness of  $k$ -nearest neighbors for grammatical error correction (kNN-GEC). Specifically, we seek to determine whether kNN-GEC always improves the performance of the base Transformer, or whether the impact varies. Additionally, we investigate how the size of the datastore affects performance, and whether synthetic data can be used to bolster the datastore. We also explore whether using synthetic data produces the same level of interpretability. Furthermore, we examine how the choice of data for the datastore impacts the effectiveness of the model on test sets. Finally, we compare the effectiveness of finetuning a Transformer on a set of data versus using that data as the datastore for kNN-GEC.

We found that the effectiveness of kNN-GEC varies depending on the base Transformer. Higher performing Transformers show little to no improvement. Using synthetic data does not appear to impact the interpretability of the corrections, and can be used to increase the size of the datastore. However, very large datastores may not improve the system’s performance enough to warrant the increase in computational cost. Bolstering the datastore with error-targeted example sentences does not seem to be a viable way of improving the system’s performance on those error types or in general. We also found that finetuning a Transformer on a set of in-domain data can be more effective than using kNN-GEC for in-domain data.

## 2 Prior Work

### 2.1 Example-based machine translation

First proposed by Nagao (1984), using examples to anchor text generation has been explored in many other tasks from summarization (Cao et al., 2018) to response generation (Weston et al., 2018). In machine translation, this process requires two steps: retrieving a relevant translation example, and using that to guide the translation of a new sentence.

Retrieving relevant example pairs is most often

done by comparing the source sentence to a datastore of source-target example pairs, and retrieving the  $k$  nearest neighbors of the source sentence. Distance may be calculated with edit-distance (Bulte and Tezcan, 2019; Hossain et al., 2020; Zhang et al., 2018), sentence embeddings (Tezcan et al., 2021; Wu et al., 2019), or a combination of both (Xu et al., 2020).

There is variety in how the retrieved example is used to produce the output sentence. Once the nearest examples are retrieved, they must be integrated into the generation. A common approach is to train a Transformer with a concatenated input of the input text and one or more retrieved target sentences (the input sentences for the translation examples are only used in retrieval) (Bulte and Tezcan, 2019; Tezcan et al., 2021; Hossain et al., 2020). This method finds the most similar examples up front, and uses a standard encoder decoder to generate the hypothesis.

Other methods involve using retrieved examples to alter the probability distribution of tokens during the autoregressive decoding. Zhang et al. (2018) proposed increasing the probabilities of the  $n$ -grams found in the output of the translation example at each timestep of decoding. Khandelwal et al. (2020) proposed an approach that could use a pre-trained seq2seq Transformer and improve its performance by retrieving examples during decoding. The nearest neighbor machine translation (kNN-MT) system uses the decoder of a pre-trained Transformer model to generate translation context vectors for each target token of the example sentences. The translation context is the source sentence and the partially generated target sentence. The vector that is passed into the final feedforward network of the decoder is considered to represent the full translation context at each time step. This vector serves as the key with the target token as the value in an example datastore of key-value pairs.

The system translates new pieces of text by consulting the datastore at each decoding step and finding the  $k$  nearest neighbors of the vector and weighting the possible output tokens by the L2 distance to the nearest neighbor keys. The authors reported significant gains using this method, especially on language pairs with a considerable number of example sentences, such as DE-EN, ZH-EN, and EN-ZH with datastore sizes of 5.56, 1.19, and 1.13 billion translation context-token pairs respectively.

## 2.2 Example-based grammatical error correction

Kaneko et al. (2022) applied kNN-MT to grammatical error correction in their EB-GEC system. The authors conducted a study using human evaluation to demonstrate that the example sentences retrieved through the decoding process improve the interpretability of the results for language learners as compared to the closest sentence pairs using edit distance or BERT-based retrieval.

EB-GEC showed mixed results compared to the vanilla Transformer model. The authors report improved performance using the  $k$  nearest neighbors at inference time on CoNLL14 (Ng et al., 2014), the test data of the BEA2019 shared task (Bryant et al., 2019), and FCE (Yannakoudakis et al., 2011), but not JFLEG (Napoles et al., 2017). As JFLEG is the only one of these test sets to focus on fluency, these results are interpreted to mean that the approach is successful at increasing accuracy of error corrections, but may not be as effective at improving the fluency of the sentence. An alternate explanation could be that the three test sets that performed better using EB-GEC had training splits that were used to train the model and also that contributed to the datastore of example sentences. JFLEG, which performed better with the vanilla model, did not have any representation in the datastore or the training.

Despite the improved accuracy on most of the test sets, EB-GEC did not perform strongly compared to the state-of-the-art tagging-based approaches to GEC. One possible reason for this may be the size of the datastore being insufficient. With only 600,000 sentences generating 17 million key-value pairs for the datastore, there may not be enough examples for the kNN system to retrieve from.

## 2.3 Synthetic examples

While most kNN-MT systems reuse bilingual training data for the datastore, it is possible that using different data or even synthetic data for translation examples could yield better results. Deguchi et al. (2022) showed that using a larger back-translated monolingual corpus for the datastore can outperform a smaller training data corpus. The reason for this has yet to be explored thoroughly. It may be simply due to the larger number of examples for the system to draw from, or it may be because the Transformer has already learned from the train-

	C4+BEA+CWEB	EB-GEC Base	PretLargeSSE
BEA-train (Bryant et al., 2019)	finetuning/dastore	training/dastore	finetuning/dastore
CWEB (Flachs et al., 2020)	finetuning	-	-
JFLEG (Napoles et al., 2017)	finetuning	-	-
gec-pseudodata (Kiyono et al., 2019)	-	-	pretraining
C4 <sub>200M</sub> (Stahlberg and Kumar, 2020)	pretraining/dastore	dastore	dastore

Table 1: How each dataset used for training, finetuning, or as the datastore was used across the three models.

ing examples and the synthetic data provides novel translation examples.

### 3 Experiments

#### 3.1 The Vanilla Transformers

In order to investigate whether the impact of kNN-GEC varies depending on the base transformer, we applied this approach to three base models trained differently on different data: C4+BEA+CWEB, EB-GEC Base, and PretLargeSSE.

C4+BEA+CWEB was trained from scratch using a combination of synthetic and authentic data with the base Transformer architecture. It was first pretrained on the synthetic corpus C4<sub>200M</sub> (Stahlberg and Kumar, 2020), which is a cleaned version of the Common Crawl. The source sentences were corrupted from the targets using a tagged seq2edits corruption method. C4+BEA+CWEB was then finetuned on BEA-train, which is composed of the training split of the First Certificate in English corpus (FCE), Lang-8 Corpus of Learner English (Tajiri et al., 2012), National University of Singapore Corpus of Learner English (NUCLE) (Dahlmeier et al., 2013), and the training split of Write & Improve + LOCNESS (Bryant et al., 2019).

C4+BEA+CWEB was also finetuned on the development split of JFLEG (Napoles et al., 2017), and CWEB (Flachs et al., 2020). JFLEG is a fluency oriented corpus which contains larger sentence edits than BEA-train. CWEB is a corpus of edits made to websites and contains much fewer and smaller sentences edits than BEA-train.

The second Transformer, EB-GEC Base was trained directly on BEA-train, with no synthetic pretraining. It was trained using the data and settings outlined in Kaneko et al. (2022)<sup>1</sup>.

The third Transformer, PretLargeSSE was pretrained on the gec-psuedodata synthetic data Kiyono et al. (2019)<sup>2</sup>. It was then fine-tuned on BEA-

<sup>1</sup><https://github.com/kanekomasahiro/eb-gec>

<sup>2</sup><https://github.com/butsugiri/gec-pseudodata>

Datastore name	Sentences	Tokens
BEA-train	1.3M	16-17M
Synthetic 2M	2M	55-78M
Synthetic 20M	20M	547-777M
Synthetic 40M	40M	1-1.6B
Synthetic Full	147M	4-7B

Table 2: The size of the datastores as measured by number of example pairs and number of resulting entries in the datastore, which is equivalent to the number of tokens of the target sentences.

train.

We provide a summary of the datasets used, which are listed in Table 1, and the detailed hyperparameters of all three base models in Appendix A.

#### 3.2 Evaluation

Testing was done on the CoNLL-2014 test data (Ng et al., 2014) using  $M^2$  (Dahlmeier and Ng, 2012), the BEA-2019 shared task test data (Bryant et al., 2019) and FCE test data (Yannakoudakis et al., 2011) using ERRANT (Bryant et al., 2017), and JFLEG (Napoles et al., 2017) using GLEU (Napoles et al., 2015).  $M^2$  and ERRANT report  $F_{0.5}$  scores.

#### 3.3 Datastores

We conducted experiments using datastores made from BEA-train and subsets of different sizes from C4<sub>200M</sub> to understand how datastore size affects performance, and whether synthetic data can improve performance. The data was preprocessed using the same method as the training data of the respective model, which leads to different sizes of each datastore depending on the vocabulary size used for subword tokenization, which varies between the three base models. The datastore ranges for the systems are noted in Table 2. C4+BEA+CWEB has a larger vocabulary size (128k), resulting in smaller datastores. The other baselines have a vocabulary size of 8k and larger datastores.

The vector that is passed into the final feedforward network of the decoder is considered to be the hidden state of the context and is used as the key

	CoNLL14	BEA2019	FCE	JFLEG
	$M^2 F_{0.5}$	ERRANT $F_{0.5}$	ERRANT $F_{0.5}$	GLEU
C4+BEA+CWEB	47.84	47.51	41.51	49.47
+ BEA-train	44.55	48.89	42.77*	48.84
+ Synthetic 2M	49.47	50.67	44.32***	50.88***
+ Synthetic 20M	51.73***	53.15	43.90***	51.83***
+ Synthetic 40M	52.09***	54.92	45.63***	51.98***
+ Synthetic Full	<b>54.17***</b>	<b>55.69</b>	<b>45.67***</b>	<b>52.49***</b>
EB-GEC Base	<b>50.01</b>	48.44	40.18	55.65
+ BEA-train	49.68	<b>51.40</b>	42.00***	<b>56.26</b>
+ Synthetic 20M	48.75	47.77	<b>42.26***</b>	52.80
PretLargeSSE	62.11	65.17	51.73	<b>60.99</b>
+ BEA-train	61.73	<b>66.44</b>	53.76***	<b>60.99</b>
+ Synthetic 2M	61.79	60.90	53.08***	59.94
+ Synthetic 20M	<b>62.46</b>	61.82	53.50***	60.15
+ Synthetic 40M	62.07	65.97	<b>53.86***</b>	60.02

Table 3: Results on test sets using  $\lambda$  of 0.5. The best result of each system is bolded.  $p$  values were calculated on CoNLL14, FCE, and JFLEG between each kNN-GEC datastore and the base model using paired bootstrap resampling.  $p < 0.05$  is denoted with \*,  $p < 0.01$  is denoted with \*\*, and  $p < 0.001$  is denoted with \*\*\*. BEA2019 test set is not released publicly, so we did not calculate the resampling for this data.

Computers **is are** the most important inventions in our **life lives**.

<i>is → are</i>	<i>invention → inventions</i>	<i>life → lives</i>
Trees <b>is are</b> the most spiritually advanced living beings on the Earth who are constantly in a <b>deflative meditative</b> state, and <b>subtle subtle</b> energy is what they speak <b>like as a</b> language.	Bitcoin <b>it is</b> one of the most important inventions <b>along in all of</b> human history.	They're the <b>earrying beginnings</b> of AI everywhere in our <b>life lives</b> .

Table 4: An example of a sentence correction and the examples used to justify each correction from the synthetic datastore Synthetic 40M.

vector in the datastore, with the target token used as the value. The datastores were indexed using FAISS (Johnson et al., 2019)<sup>3</sup>, with a training size of 5,242,880 and a chunk size of 10,000,000. Product quantization was applied to split the vectors into 64 subspaces and quantize each subspace. In addition, the vectors were clustered using k-means clustering into 131,072 clusters to speed up search.

During decoding, the  $k$  nearest vectors to the hidden state passed to the final feedforward network in the decoder are retrieved from the datastore. In this work,  $k$  is set to 16 and "nearest" is defined by shortest Euclidean distance.

Using the full synthetic datastore is computationally expensive, and the results on our initial experiments show marginal improvement from the next smaller datastore (40M). For this reason, the results of using the full synthetic datastore for kNN-GEC were not calculated for the other two systems. The results of each system are listed in Table 3.

Synthetic data was most effective when used with C4+BEA+CWEB. Even the smallest synthetic

datastore improved the score compared to the non-synthetic datastore, and larger datastores resulted in even higher scores. However, the same gains were not seen by PretLargeSSE, which was the highest scoring base model. PretLargeSSE showed mixed results with the authentic datastore, and the synthetic datastore was less effective. It took at least 20M sentences for the synthetic datastore to perform as well as the much smaller authentic one.

### 3.4 Interpretability

One advantage of kNN-GEC is that it can provide motivating examples from the datastore for the corrections it makes. Kaneko et al. (2022) showed that the examples sentence pairs used during decoding provided more relevant models for learners than those retrieved by word overlap or BERT embeddings. As we investigate the effectiveness of using synthetic sentences in the datastore to improve the quality of model corrections, it is important to ensure that synthetic sentences can also serve as effective models for learners. To accomplish this, our code generates the kNN examples for each to-

<sup>3</sup><https://github.com/facebookresearch/faiss>



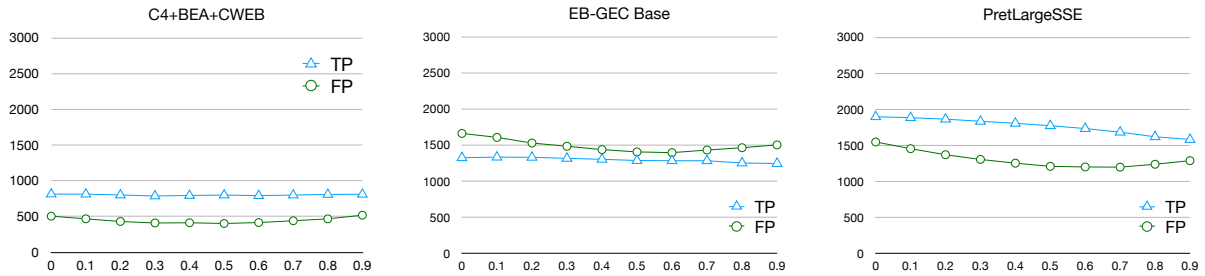


Figure 1: The number of true positives and false positives in the FCE test set for different values of  $\lambda$  using the BEA-train dataset.

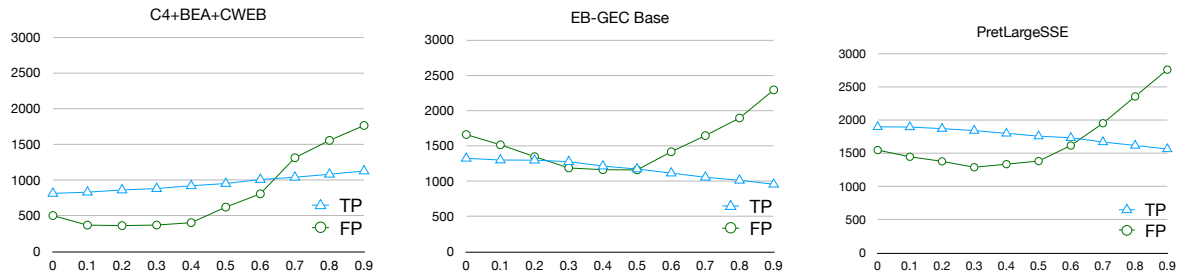


Figure 2: The number of true positives and false positives in the FCE test set for different values of  $\lambda$  using the Synthetic 20M dataset.

ken in the corrected sentence. In a post-processing step, we align the source sentence and corrected sentence using ERRANT (Bryant et al., 2017) and extract the nearest example for each corrected token. To explore the interpretability of the synthetic dataset, we present a randomly selected sentence correction with motivating examples in Table 4.

In this single example, the synthetic dataset provides reasonable examples for three corrections. A larger study measuring the effectiveness of synthetic data on the quality of the examples provided by the model is left for future work.

### 3.5 Impact of kNN on corrections

To investigate the effects of kNN-GEC on the models, we adjusted the hyperparameter  $\lambda$ . This parameter regulates the proportion of the probability distribution for the next token that comes from the dataset. kNN-GEC uses a linear interpolation between the output of the kNN token distribution,  $p_{\text{kNN}}$ , and the vanilla decoder,  $p_{\text{GEC}}$ , as follows:

$$P(y_i|x, y_{1:i-1}) = \lambda p_{\text{kNN}}(y_i|x, y_{1:i-1}) + (1 - \lambda) p_{\text{GEC}}(y_i|x, y_{1:i-1}) \quad (1)$$

The hyperparameter  $\lambda$  in equation 1 is used to balance the probability distribution generated by the example sentences and that generated by the base Transformer. Setting  $\lambda$  to 0 is equivalent to using the vanilla Transformer without kNN-GEC. The

larger the  $\lambda$ , the more the system will use the retrieved examples when generating the next token. However, using only the examples can lead to errors, so we did not calculate  $\lambda$  of 1. Figure 1 shows the number of true positives and false positives generated in the FCE test data by each system using the BEA-train dataset.

In general, the use of kNN-GEC does not increase the number of corrections made until  $\lambda$  values exceed 0.5. In all three systems, using the BEA-train dataset leads to a more conservative approach to corrections, which results in fewer incorrect changes being made. One possible explanation is that the method of retrieving example sentence pairs returns pairs that have similar meanings or are on similar topics but may not necessarily contain the same errors. In the absence of an error, GEC will copy from the input sentence to the hypothesis sentence. If the  $k$  retrieved sentence pairs do not contain the error, kNN-GEC may copy more and correct less.

Figure 1 and Figure 2 show how  $\lambda$  affects the number of true positives and false positives in the FCE test set. These figures show that the gain in  $F_{0.5}$  score below  $\lambda = 0.5$  is due to an increase in precision resulting from a decrease in false positives, rather than an increase in true positives. While the number of true positives does not decrease rapidly, the number of false positives does, leading to better model performance despite the decrease in recall.

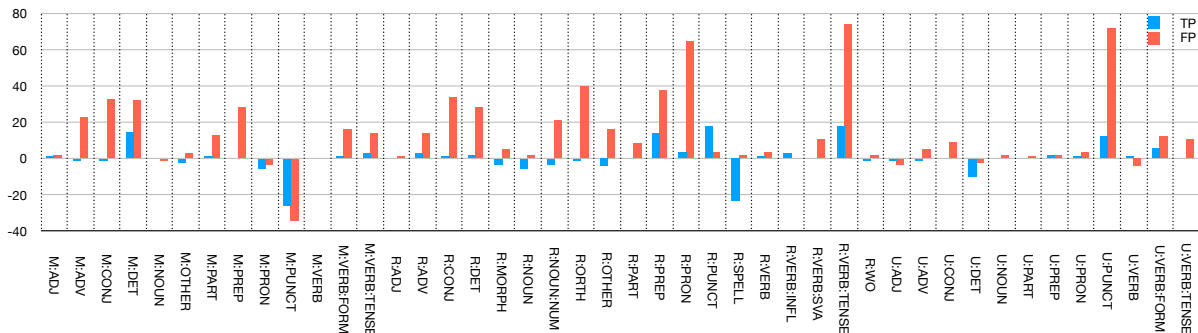


Figure 3: The difference in the number of true positives (TP) and false positives (FP) generated using the corresponding targeted datastore compared to the vanilla Transformer.

Interestingly, the number of true positives increases when using the synthetic datastore with C4+BEA+CWEB, as shown in figure 2. However, this effect doesn’t transfer to PretLargeSSE. This may be because C4+BEA+CWEB is already a very conservative model and performs poorly in comparison to PretLargeSSE. C4+BEA+CWEB may not be as effective at producing corrections as PretLargeSSE, and having more example sentences to use improves the results. Further work is needed to determine the reason for this difference.

All systems exhibit an increase in false positives when  $\lambda > 0.5$ , with the synthetic datastore demonstrating this most dramatically. Generally, using  $\lambda = 0.4$  resulted in the best balance of precision and recall. However, as many papers use 0.5 as the balancing point between the kNN distribution and the vanilla Transformer, the rest of the experiments in this work use  $\lambda = 0.5$ .

### 3.6 Error type targeted datastores

To observe how changing the error distribution in the datastore impacts the effectiveness of the model on that error type, as well as the performance of the system as a whole, we conducted experiments using datastores that contained examples with a single error type. We extracted 10,000 sentences from synthetic data for most<sup>4</sup> of the ERRANT error tags. The ERRANT error tags consist of an error category and error type. The error categories are Missing (M), Replacement (R), and Unnecessary (U). The error types include Adjective (ADJ), Adverb (ADV), Morphology (MORPH), Orthography (ORTH), and more. There are a total of 54 error tags, of which 8 didn’t have enough data to generate a targeted datastore. For each of the remaining 46

<sup>4</sup>Some errors were very rare and did not occur more than a handful of times in the data.

Datastore	TP	FP	P	R	$F_{0.5}$
None	1,896	1,548	55.05	41.68	51.73
BEA	1,819	1,273	58.83	39.99	53.76
+30K	1,355	921	59.53	29.79	49.62

Table 5: A comparison of different datastores, BEA-train+30K includes 10K synthetic example pairs each of Missing Adjectives, Missing Particles, and Replacing Verb Inflections corrections

error tags, a datastore was constructed with 10,000 sentences that contained only that error. We used the target datastore with PretLargeSSE and tested it on FCE-test with a  $\lambda$  value of 0.5.

Using much smaller targeted datastores alone lowers both precision and recall compared to the base Transformer and performs much worse than the BEA-train datastore. Instead of looking at the overall performance, we examine how the number of true positives and false positives changes within the targeted error type compared to the base Transformer. Figure 3 illustrates the difference in the number of true or false positives between using the targeted datastore and the vanilla Transformer.

Using an error-targeted datastore tends to increase the number of false positives for a particular error, likely due to the system overapplying the correction. Surprisingly, for many error types, the targeted datastore does not increase the number of true positives. However, it does increase the accuracy of correcting missing determiners (M:DET), incorrect prepositions (R:PREP), verb tense (R:VERB:TENSE), and unnecessary punctuation (U:PUNCT). The number of false positives often increases much more than the number of true positives, resulting in lower precision. Replacing incorrect punctuation (R:PUNCT) is an exception, as it can increase precision without significantly increasing false positives.

It is reasonable to assume that a datastore con-

	CoNLL14	BEA2019	FCE	JFLEG
Finetuned	<b>62.11</b>	<b>65.17</b>	<b>51.73</b>	60.99
kNN-GEC	56.34	58.58	47.06	<b>61.92</b>

Table 6: Results of finetuning a Transformer on BEA-train compared to using kNN-GEC with BEA-train as the datastore.

taining only one type of error will perform poorly on a test set with many diverse errors, since there are no examples for any of the other types of corrections to use as a model. However it’s not immediately obvious if targeted datastores could be used to supplement a base datastore in order to compensate for low performance with a particular error type. Preliminary experiments suggest that this approach may not be effective for all error types. Table 5 presents the results of adding 30,000 synthetic pairs to the BEA-train datastore to address three types of low-performing errors: missing adjectives, missing particles, and incorrect verb inflections. These three types were selected because using kNN-GEC with the BEA-train datastore alone did not produce any more true positives than the base Transformer, but each of these three types saw a slight increase in true positives in their respective categories when the targeted datastore was used.

Adding the 30,000 new sentence pairs decreased the number of false positives, but it also significantly reduced the number of true positives, making the overall system more conservative. This resulted in an increase in precision. Unfortunately, the decrease in recall lowered the overall  $F_{0.5}$  score to less than that of the vanilla Transformer.

### 3.7 Finetuning vs kNN-GEC

To determine the effectiveness of finetuning a Transformer on data versus using that data as a kNN-GEC datastore, we tested using a checkpoint of PretLargeSSE before the finetuning phase (Kiyono et al., 2019). We applied the kNN-GEC method with this pretrained-only model, using BEA-train as the datastore. The results of this experiment are shown in Table 6.

The Transformer model that was finetuned outperformed the pretrained-only model using kNN-GEC on three of the four test sets. The three datasets that performed better with finetuning (CoNLL14, BEA2019, and FCE) all have training sets used in the finetuning or kNN-GEC. This suggests that finetuning is more effective for in-domain test sets.

## 4 Discussion

Overall, kNN-GEC makes a base GEC system more conservative about making corrections, which lowers its recall. This is likely due to the fact that the chosen examples may not contain a similar error to the one being corrected, but are closer in content to the example pair. The retrieval method involves comparing embedded vectors from the decoder, which contain a mixture of information about the syntax and semantics. As a result, there can be times when the closest sentence pair to the one being decoded overlaps more heavily on semantics than syntax. It is likely that the tendency to retrieve example sentence pairs that are similar in content but not grammatically incorrect promotes more copying, or more conservative corrections, as the target word may not even be incorrect in the example pair.

This is a key difference between the tasks of machine translation and grammatical error correction. Machine translation must generate the appropriate content words for the translation, while GEC mostly uses the content words from the source sentence, or a different form of the existing word. MT may benefit from the influence of the kNN probability distribution on word choice because success in MT often includes selecting the correct content word for the context. In the case of GEC, however, example sentence pairs may not contain the same grammatical errors as the query sentence, which leads to more copying and less correcting.

## 5 Conclusion

In this work, we investigate how using training examples during decoding with the kNN-GEC method affects the precision and recall of grammatical error correction. We used three different base models and found that the effectiveness of kNN-GEC varies greatly depending on the base model. In general, this method makes models more conservative in making corrections, improving precision but lowering recall. Synthetic data can be used to increase the size of the datastore, but its effectiveness depends on the base model. While kNN-GEC using authentic or synthetic datastores increases the interpretability of corrections for learners, this comes with the trade-off of fewer corrections made and a longer decoding time. We also explored the effect of the hyperparameter  $\lambda$  on the performance of the kNN-GEC method. A value of 0.4 tended to produce the best balance of precision and recall,

though many papers use a value of 0.5. Finally, we compared the effectiveness of finetuning a Transformer on a set of data versus using that data as the datastore for kNN-GEC. Our results showed that finetuning the Transformer on the data generally outperformed using the data as a datastore.

Given that the more conservative corrections indicate that kNN-GEC is retrieving examples that are more semantically similar than those containing similar corrections, a future direction for kNN-GEC could be selecting the  $k$  nearest neighbors based on similarity of errors rather than similarity of content. This would require a target output that expresses the difference between the input sentence and the hypothesis sentence. During decoding, the necessary edits would be applied to change the example source to the example target token to the input. Future work could involve separating the syntax from the semantics of the encoded input sentence to retrieve the nearest neighbors with syntactically similar example sentences. This would help overcome the limitations of the kNN system with regards to making new corrections.

## Limitations

The three base models used for the experiments were trained with different settings. As a result, it is challenging to understand the exact source of discrepancies between the results. Additionally, each of the three models used different subword tokenizations, resulting in variable datastore sizes. Although we have some hypotheses about why kNN affects GEC differently from MT, more experiments need to be conducted to confirm them.

## References

- Chris Brockett, William B. Dolan, and Michael Gamon. 2006. [Correcting ESL errors using phrasal SMT techniques](#). In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 249–256, Sydney, Australia. Association for Computational Linguistics.
- Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. [The BEA-2019 shared task on grammatical error correction](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.
- Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. [Automatic annotation and evaluation of error types for grammatical error correction](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.
- Bram Bulte and Arda Tezcan. 2019. [Neural fuzzy repair: Integrating fuzzy matches into neural machine translation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1800–1809, Florence, Italy. Association for Computational Linguistics.
- Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2018. [Retrieve, rerank and rewrite: Soft template based neural summarization](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 152–161, Melbourne, Australia. Association for Computational Linguistics.
- Daniel Dahlmeier and Hwee Tou Ng. 2012. [Better evaluation for grammatical error correction](#). In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572, Montréal, Canada. Association for Computational Linguistics.
- Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. [Building a large annotated corpus of learner English: The NUS corpus of learner English](#). In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31, Atlanta, Georgia. Association for Computational Linguistics.
- Hiroyuki Deguchi, Kenji Imamura, Masahiro Kaneko, Yuto Nishida, Yusuke Sakai, Justin Vasselli, Huy Hien Vu, and Taro Watanabe. 2022. [Naist-nicttit wmt22 general mt task submission](#). In *Proceedings of the Seventh Conference on Machine Translation*, pages 244–250, Abu Dhabi. Association for Computational Linguistics.
- Simon Flachs, Ophélie Lacroix, Helen Yannakoudakis, Marek Rei, and Anders Søgaard. 2020. [Grammatical error correction in low error density domains: A new benchmark and analyses](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8467–8478, Online. Association for Computational Linguistics.
- Nabil Hossain, Marjan Ghazvininejad, and Luke Zettlemoyer. 2020. [Simple and effective retrieve-edit-rerank text generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2532–2538, Online. Association for Computational Linguistics.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

- Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. [Approaching neural grammatical error correction as a low-resource machine translation task](#). 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 595–606, New Orleans, Louisiana. Association for Computational Linguistics.
- Masahiro Kaneko, Sho Takase, Ayana Niwa, and Naoaki Okazaki. 2022. [Interpretability for language learners using example-based grammatical error correction](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7176–7187, Dublin, Ireland. Association for Computational Linguistics.
- Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. [Nearest neighbor machine translation](#). *CoRR*, abs/2010.00710.
- Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. [An empirical study of incorporating pseudo data into grammatical error correction](#). 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 1236–1242, Hong Kong, China. Association for Computational Linguistics.
- Makoto Nagao. 1984. A framework of a mechanical translation between Japanese and English by analogy principle.
- Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. [Ground truth for grammatical error correction metrics](#). 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 588–593, Beijing, China. Association for Computational Linguistics.
- Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. [JFLEG: A fluency corpus and benchmark for grammatical error correction](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 229–234, Valencia, Spain. Association for Computational Linguistics.
- Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. [The CoNLL-2014 shared task on grammatical error correction](#). In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.
- Hiroki Ouchi, Jun Suzuki, Sosuke Kobayashi, Sho Yokoi, Tatsuki Kuribayashi, Ryuto Konno, and Kentaro Inui. 2020. [Instance-based learning of span representations: A case study through named entity recognition](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6452–6459, Online. Association for Computational Linguistics.
- Felix Stahlberg and Shankar Kumar. 2020. [Seq2Edits: Sequence transduction using span-level edit operations](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5147–5159, Online. Association for Computational Linguistics.
- Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. [Tense and aspect error correction for ESL learners using global context](#). In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 198–202, Jeju Island, Korea. Association for Computational Linguistics.
- Arda Tezcan, Bram Bulté, and Bram Vanroy. 2021. [Towards a better integration of fuzzy matches in neural machine translation through data augmentation](#). *Informatics*, 8(1).
- Jason Weston, Emily Dinan, and Alexander Miller. 2018. [Retrieve and refine: Improved sequence generation models for dialogue](#). In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pages 87–92, Brussels, Belgium. Association for Computational Linguistics.
- Jiawei Wu, Xin Wang, and William Yang Wang. 2019. [Extract and edit: An alternative to back-translation for unsupervised neural 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 1173–1183, Minneapolis, Minnesota. Association for Computational Linguistics.
- Jitao Xu, Josep Crego, and Jean Senellart. 2020. [Boosting neural machine translation with similar translations](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1580–1590, Online. Association for Computational Linguistics.
- Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. [A new dataset and method for automatically grading ESOL texts](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.
- Jingyi Zhang, Masao Utiyama, Eiichiro Sumita, Graham Neubig, and Satoshi Nakamura. 2018. [Guiding neural machine translation with retrieved translation pieces](#). 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 1325–1335,

New Orleans, Louisiana. Association for Computational Linguistics.

## A Hyperparameters

The detailed hyperparameters of the base Transformers and the settings used for generation.

	<b>C4+BEA+CWEB</b>	<b>EB-GEC Base</b>	<b>PretLargeSSE</b>
Architecture	Transformer Base	Transformer Big	Transformer Big
Optimizer	Adam ( $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1 \times 10^{-8}$ )		Pretrained with Adam, Fine-tuned with Adafactor
Learning Rate Schedule	Inverse square root decay	Inverse square root decay	Fixed
Warmup Steps	4,000	4,000	-
Dropout	0.2	0.3	0.1
FFN size	2096	4096	4096
Gradient Clipping	1.0	0.0	0.0
Label Smoothing	$\epsilon_{ls} = 0.1$	$\epsilon_{ls} = 0.1$	None
Layers	Encoder 6, Decoder 4	Encoder 6, Decoder 6	Encoder 6, Decoder 6
Mini-batch Size	4096 tokens	4096 tokens	unknown
Number of Updates	10,800 steps	20 epochs	unknown

Table 7: Hyperparameters of the vanilla Transformers.

<b>Generation settings</b>	
Length Penalty	1.0
Beam Size	5
Temperature	100
$\lambda$	0.5

Table 8: Settings for the kNN generation

# Towards Extracting and Understanding the Implicit Rubrics of Transformer Based Automated Essay Scoring Models

**James Fiacco**

Language Technologies Institute  
Carnegie Mellon University  
jfiacco@cs.cmu.edu

**David Adamson**

Turnitin  
dadamson@turnitin.com

**Carolyn P. Rosé**

Language Technologies Institute  
Carnegie Mellon University  
cprose@cs.cmu.edu

## Abstract

By aligning the functional components derived from the activations of transformer models trained for AES with external knowledge such as human-understandable feature groups, the proposed method improves the interpretability of a Longformer Automated Essay Scoring (AES) system and provides tools for performing such analyses on further neural AES systems. The analysis focuses on models trained to score essays based on ORGANIZATION, MAIN IDEA, SUPPORT, and LANGUAGE. The findings provide insights into the models' decision-making processes, biases, and limitations, contributing to the development of more transparent and reliable AES systems.

## 1 Introduction

Since its inception over 50 years ago (Page, 1966), Automated Essay Scoring (AES) has been a valuable approach for evaluating large quantities of student essays. Recent developments in the field have sought to harness advanced natural language processing techniques to score essays on par with human raters, achieving significant progress toward that goal (Ramesh and Sanampudi, 2022; Huawei and Aryadoust, 2023; Mizumoto and Eguchi, 2023). The inability to understand the learned representations in deep learning based AES models introduces risk and validity concerns to their widespread use in educational settings (Ding et al., 2020; Kumar et al., 2020, 2023). In response to this concern, we propose a functional component-based approach to scrutinize the activations of transformer models trained for AES.

The primary goal of this study is to provide a method and tool that can provide a coherent and interpretable understanding of the functions per-

formed by these neural models, comparing their overlaps and differences, and aligning the learned functions with human-understandable groups of features<sup>1</sup>. Much in the same way that human evaluators use rubrics to guide their scoring of essays, neural models learn a set of features and connections that, when combined and applied to an essay, repeatedly determine the score that they will assign. Through the comparison and contrast of these components across models, we investigate how the models prioritize different aspects of writing and make stride towards unveiling that their learned rubrics are, alongside any underlying biases or limitations that they entail. Ultimately, this in-depth analysis will enhance our understanding of the neural models' decision-making processes, thereby contributing to the development of more transparent and reliable automated essay scoring systems.

Our proposed methodology involves extending the emerging domain of neural network interpretation by using abstract functional components, enabling a robust comparison between probed functional components of a network and independent feature groups. This approach specifically builds upon recent work on neural probes and derived methods, aligning a neural network's activations with external knowledge such as task metadata and implicit features (e.g., parts-of-speech, capitalization, etc.) (Conneau et al., 2018; Belinkov, 2022). We focus our interpretation in the domain of AES where each model in our investigation is trained to score essays based on distinct evaluation traits, namely ORGANIZATION, MAIN IDEA, SUPPORT, and LANGUAGE.

To probe these models, the features are drawn

<sup>1</sup>Code and tool available at [https://github.com/jfiacco/aes\\_neural\\_functional\\_groups](https://github.com/jfiacco/aes_neural_functional_groups)



from several sources that correspond to concepts of both high and low validity for essay scoring: statistical features of an essay (e.g. number of sentences, number of paragraphs, etc.) (Woods et al., 2017), tree features generated from Rhetorical Structure Theory (RST) (Mann and Thompson, 1987) parses of the essays (Jiang et al., 2019; Fiacco et al., 2022), essay prompt and genre (West-Smith et al., 2018), and a combination of algorithmically derived (Derczynski et al., 2015) and our own human defined style-based word lists. These features provide a lens that while unable to capture all of the capabilities of the models, provide insight into some of the key differences between them.

In the following sections, we provide a detailed description of the methodology used for this analysis, discuss the assumptions underpinning the method, and present potential explanations for correlated function/feature pairs through a series of experiments that validate our method’s ability to reflect the internal rubric of each of the neural models.

## 2 Related Work

From the interpretability angle, the most closely related work to this is that of neural model probes (Shi et al., 2016; Adi et al., 2016; Conneau et al., 2018; Zhu et al., 2018; Kuncoro et al., 2018; Khandelwal et al., 2018) which have frequently been used to test whether a model has learned a set of properties (Ryskina and Knight, 2021; Belinkov, 2022). The primary gap we are working to fill in from this body of literature is that current approaches, with few exceptions (Fiacco et al., 2019; Cao et al., 2021), focus on understanding the roles of individual neurons in the greater neural network. We contend that studying the interpretability of a neural network at the individual neuron level can too easily obscure the broader picture. Our interest lies in further progress incorporating a more abstract perspective on what is learned by neural networks, complementing the work that has been done at the neuron level.

Compared to alternative paradigms for interpretability in machine learning models, such as LIME (Ribeiro et al., 2016) or SHAP (Lundberg and Lee, 2017), which evaluate the contribution of a given feature to the prediction of a model, the functional component based methods allow for a more granular identification of important parts of a model, independent from known features for a

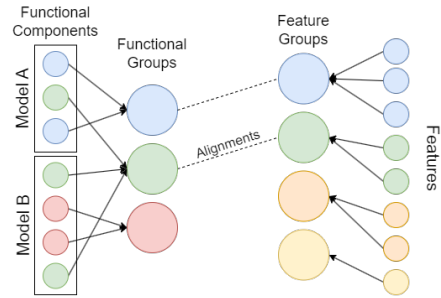


Figure 1: Diagram visualizing the structure of the methodology. Nodes of each color represent correlated values.

task. This can enable model analysts to quickly identify unexplained components and begin to propose alternative pallets of features. Furthermore, the functional components can represent intermediate steps within the neural network which would be unobservable with these alternative methods.

From the educational technologies and Automated Essay Scoring angle, our work primarily applies to the body of deep learning-based AES models such as recurrent neural network models (Jin et al., 2018; Nadeem et al., 2019), convolutional neural network models (Taghipour and Ng, 2016), and transformer models (Sethi and Singh, 2022). While our method could be applied to any type of neural model, we focus on transformers as they represent the state-of-the-art. By integrating the interpretability of neural models with the understanding of the functional components they learn, we hope to bridge the gap between human-understandable features and neural network-based essay scoring. The insights gained from our methodology can guide the development of more effective and efficient AES systems, tailored to the specific needs of educators and students. Furthermore, the lessons learned from this research may extend beyond the AES domain, providing valuable insights for the broader field of natural language processing and machine learning interpretability.

## 3 Methods

In this section we present our interpretation approach (Figure 1), defining the key concepts of *functional components*, *functional group*, *feature*, and *feature group*. Because the approach notably abstracts away from common terms in the neural network literature, throughout this section we draw an analogy to how one can define and describe the common features between mammals by comparing

their common and unique characteristics.

### 3.1 Functional Components and Groups

Functional components refer to the learned functions of a neural network, much like a particular component of a dog may be a “dog leg”. In a neural AES system, these would be a group of neurons that have correlated activations when varying the input essays. The approach to extracting functional components (“neural pathways” as described by [Fiacco et al. \(2019\)](#)) from a neural network consists of finding the sets of coordinated neuron activations, summarized by the following steps:

1. Save the activations of neurons for each data instance in the validation dataset into an activation matrix,  $A$  of size  $M \times N$ , where  $M$  is the number of data instances in the validation set and  $N$  is the number of neurons being used for the analysis.
2. Perform a dimensionality reduction, such as Principal Component Analysis (PCA) ([Hotelling, 1933](#)), on  $A$  to get component activation matrix,  $T_{model}$  of size  $M \times P$ , where  $P$  is the number of principal components for a given model.

Functional groups are collections of similar functional components. Continuing the analogy, they would be compared to the more general concept of a “leg”. We compute functional groups by concatenating the dimensionality reduced matrixes,  $T_{model}$ , of the two models that are to be compared and performing an additional dimensionality reduction over that matrix to get a matrix of group activations,  $T$ . The functional components that are highly loaded onto each functional groups are considered members of that group. An important departure from [Fiacco et al. \(2019\)](#), stemming from the limitation that does PCA does not guarantee independence between components, is that we use Independent Component Analysis (ICA) ([Comon, 1994](#)) instead. ICA is a dimensionality reduction technique that maximizes the independence between components, resulting in more validity in the technique’s resulting alignments.

To determine if a functional group is influential in the performance of the model (designating it an *important functional group*), we can compute the Pearson’s correlation coefficient between each column of the group activation matrix and the pre-

dictions of the model, the errors of the model, and the differences between the compared models.

### 3.2 Independent Feature Groups

Features are human understandable attributes that can be extracted from an analysis dataset. In the analogy they would represent potential descriptors of a components of a mammal, e.g. “hairy”. In an AES context, these features may manifest as “no capitalization after a period”. Ideally, it would be possible to create a direct mapping from each of the functional components to each of the features for which the functional component is related. However, this is non-trivial during a post-hoc analysis because, without interventions, there are limitations on what information is obtainable. Specifically, because features are not necessarily independent from each other, their correlations cannot be separated from each other, yielding imprecise interpretations. It is thus required for only independent features to be used as the unit of analysis when it comes to alignment with functional components. Unfortunately, in practice, this is a prohibitive restriction and most features that would be interesting are going to have correlations.

Fortunately, much in the same way that we can use ICA to extract independent functional components from a neural network’s activations, we can use it to construct independent feature groups that can be reasonably be aligned with the functional groups of the neural networks. In the analogy, these independent feature groups can therefore, be thought of as collections of descriptive terms that can identify a characteristic of the mammal, such as “an appendage that comes in pairs and can be walked on” which would align with the “leg” functional group. In AES, an example feature group may be “uses punctuation improperly”. It would be expected that this feature group would align well with a functional group in a neural AES system that corresponds with a negative essay score. Furthermore, feature groups for AES can be thought of as being roughly analogous to conditions that would be on an essay scoring rubric (as well as potentially other features that may be intuitive or obvious to human scorers but contribute to accurate scoring).

The specific process used to define these groups is to perform a dimensionality reduction on each set of feature types that may have significant correlations and collecting them into a feature matrix. We do this process for each feature type rather than

over all features at once because spurious correlations between some unrelated features may convolute the feature groups, making them far more difficult to interpret.

### 3.3 Alignment

Using ICA as the dimensionality reduction, the independent functional groups of the neural model can reasonably align with the independent feature groups using the following formal procedure: given a neural network,  $N$ , with activation matrix,  $A$  (as above), an independent component analysis is performed yielding a set of functional components,  $F$ . For each  $f_i, f_k \in F$ ,  $f_i \perp\!\!\!\perp f_k | X, Y$ , where  $X$  is the set of inputs to the neural network and  $Y$  is the set of predictions from the neural network. With a sufficient number of components such that  $F$  contains all independent functional components in  $A$ , if there exists a common latent variable in both  $N$  and the set of independent feature groups,  $G$ , with components  $g_i \in G$ , then there will be some  $f_i \approx g_j$ .

## 4 Experiments

In this section, we delve into the specific methodology used to analyze the activations of the four transformer models for AES, as well as the steps taken to prepare the data and features for this analysis.

### 4.1 Datasets

Although scoring rubrics are specific to the genre and grade level of a writing task, there are commonalities between each rubric that allow their traits to be reasonably combined for modeling. All our rubrics, for example, include LANGUAGE (and style) and ORGANIZATION traits, though their expectations vary by genre and grade level. The generic MAIN IDEA trait corresponds to “Claim” and “Clarity and Focus” traits, and SUPPORT corresponds to “Support and Development” as well as “Analysis and Evidence.” Rubrics and prompts were developed for validity, and essays were rigorously hand-scored by independent raters in the same manner as described in West-Smith et al. (2018).

For each generic trait, the training set was sampled down from over 50,000 available essays, responding to 95 writing prompts. Essays from 77 prompts were selected for the training set, and another 18 were held out for evaluation. Within

each split, essays were sampled to minimize imbalance between essay score, genre, grade level. In the un-sampled data, longer essays tend to be strongly correlated with essay score, risking overfitting to this surface feature. Similarly, among the subset of data where school district data was available, districts with predominantly Black enrollment were under-represented among essays with a score of “4” across all traits. To counteract these potential biases, the available data was binned by length and district demographic information for each score, genre, and grade level, and essays were under-sampled from the largest bins. In addition to these balanced essays, about 800 “off topic” essays representing nonsense language or non-academic writing were included in the dataset, with a score of zero.

### 4.2 Models

Longformers are a transformer-based neural network architecture that have gained prominence in various NLP tasks (Beltagy et al., 2020). In the context of AES, each generic trait’s model is a Longformer with a single-output regression head, fine-tuned on the trait’s balanced dataset: For the remainder of this paper, the model fine-tuned on a given trait will be referred to as “the TRAIT model” (e.g. the ORGANIZATION model) for simplicity.

Although ordinal scores from 0 to 4 were used for sampling and evaluation, the training data labels were continuous, averaged from rater scores. Essays were prefixed with text representing their genre (e.g., “Historical Analysis”) and prompt’s grade range (e.g., “grades 10-12”) before tokenization, but no other context for the writing task (e.g., the prompt’s title, instructions, or source material) was included. In addition to Longformer’s sliding attention window of 512 tokens, the first and last 32 tokens received global attention.

Scores were rounded back to integers between 0 and 4, before evaluation. On the holdout prompts, overall Quadratic Weighted Kappa (QWK) ranged from 0.784 for MAIN IDEA to 0.839 for LANGUAGE, while correlation with word count remained acceptably low: 0.441 for LANGUAGE up to 0.550 for SUPPORT.

The activations of the Longformer model were saved for each instance in the analysis set at the “classify” token to create a matrix of activations for the functional component extraction.

Model A	Model B	# Essays	Extracted Features	# Independent Feature Groups	# Aligned IFG
ORGANIZATION	MAIN IDEA	407	148	114	24
ORGANIZATION	LANGUAGE	275	118	86	39
ORGANIZATION	SUPPORT	144	90	63	37
LANGUAGE	MAIN IDEA	341	129	95	26
LANGUAGE	SUPPORT	72	67	38	23
SUPPORT	MAIN IDEA	260	127	94	27

Table 1: Comparing analysis dataset size and numbers of extracted features for each of the model comparisons, identified by the Model A and Model B columns.

### 4.3 Features

The features employed in this analysis encompass statistical properties of the essays, tree features generated from Rhetorical Structure Theory (RST) parse trees of the essays, essay prompt and genre, a combination of algorithmically derived and human-defined style-based word lists, and certain school-level demographic features. A description of each feature type is provided below:

**Statistical Features:** While statistical features such as *essay word count* are often good indicators of essay score, they are not intrinsically valuable to the different traits that our models are scoring. We thus want to see lower alignment with these features to indicate that the model is not overly relying on rudimentary shortcuts scoring an essay. We also include *average word length*, *essay paragraph count*, *essay sentence count*, *average sentence length*, and the *standard deviation of the sentence length* for completeness.

**RST Tree Features:** These features were integrated to capture the rhetorical structure of the text, such as the hierarchy of principal and subordinate clauses, the logical and temporal relations between propositions, and the coherence of the argument. These concepts have a high validity for scoring essays (Jiang et al., 2019), especially for ORGANIZATION, so high alignment between functional groups would be expected. To generate RST trees for each essay, we utilize a pretrained RST parser specifically fine-tuned for student writing (Fiacco et al., 2022). We include the presence of an RST relation as a feature as well as relation triplets ( $REL_{parent}$ ,  $REL_{child_1}$ ,  $REL_{child_2}$ ) as tree-equivalent n-gram-like features.

**Essay Prompt and Genre:** Categorical representations of the essay prompt and genre were employed as features to examine if components of the AES model were preferentially activated based on the content or topic of the essay, a low validity feature.

#### Algorithmically Generated Word List Features:

We calculate the frequency of usage of words within algorithmically derived sets of words in the essays as a group of features to probe the AES model’s consideration for stylistic language. To generate these word lists, we obtain Brown clusters (Brown et al., 1992) from essays. We generate separate Brown clusters for each prompt in our dataset and subsequently derive final word lists based on the overlaps of those clusters. This approach emphasizes common stylistic features as opposed to content-based clusters.

**Human Generated Word List Features:** In addition to the algorithmically defined word lists, we devise our own word lists that may reflect how the AES model scores essays. We created word lists for the following categories: simple words, informal language, formal language, literary terms, transition words, and words unique to African American Vernacular English (AAVE).

**Demographic Features:** We used the percent to participants in the National School Lunch Program (NSLP) at a school as a weak proxy for the economic status of a student. Also as weak proxies for economic status of essay authors, we include the school level features of *number of students* and *student teacher ratio*. Furthermore, we use a school level distribution of ethnicity statistics as a weak proxy for the ethnic information of an essay’s author. These features were employed to investigate the model’s perception of any relationship between the writer’s background and the quality, content, and style of the essay, in order to gain insight of the equity of the AES model.

### 4.4 Analysis Settings

To choose the number of components for ICA, a PCA was performed to determine how many components explained 95% of the variance of the activation (or 99% of the variance for the features) to be used as the number of components of the ICA.

Model A	Model B	Functional Group Extraction			Important Functional Group Alignment			
		# Comp. A	# Comp. B	# FG	# Aligned FG	# A Only	# B Only	# Mixed
ORGANIZATION	MAIN IDEA	119	55	125	22	12	0	10
ORGANIZATION	LANGUAGE	96	66	110	29	11	0	18
ORGANIZATION	SUPPORT	66	36	68	22	9	1	12
LANGUAGE	MAIN IDEA	78	55	93	23	8	3	12
LANGUAGE	SUPPORT	34	28	38	13	2	2	9
SUPPORT	MAIN IDEA	45	49	64	25	2	2	21

Table 2: Comparing number of functional groups extracted for each model comparison and presenting the number of functional groups that were both deemed important (Section 3.1) and sufficiently aligned with at least one feature group. Also specified is the number of functional groups that are unique to a particular model and the number that are shared between the models of given a comparison pair.

To determine that a functional group was important, it needed to have an absolute value of Pearson’s  $r$  value of greater than 0.2. This threshold was also used to determine if a functional group should be considered aligned with a feature group.

## 5 Results

In this section, we present aggregate statistics for each model comparison when it comes to computing features and independent feature groups (Table 1), extracting functional groups and aligning important functional groups (Table 2), and lastly, we provide examples taken from the model comparison between the LANGUAGE model and the MAIN IDEA model. Due to length constraints, we present detailed examples of this comparison only. Similar figures and correlation statistics can be found on Github<sup>2</sup>.

### 5.1 Independent Feature Groups

Since each trained model held out a different set of prompts from its training set, common prompts between analysis sets needed to be identified, and thus the number of features extracted and the resulting independent feature groups vary between model comparisons. Computing the independent feature groups for each model comparison (Table 1) yielded between 70% and 77% of the original extracted features for all comparisons, except LANGUAGE v SUPPORT, which only yielded 57% as many independent feature groups compared to original features. Despite high variability in the number of independent feature groups identified during the process, a much more narrow range of independent feature groups was aligned during the analysis.

<sup>2</sup>[https://github.com/jfiacco/aes\\_neural\\_functional\\_groups/tree/main/supplementary\\_results](https://github.com/jfiacco/aes_neural_functional_groups/tree/main/supplementary_results)

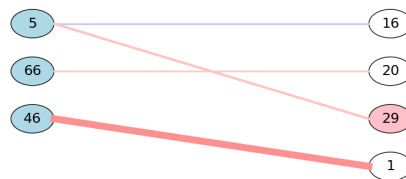


Figure 2: Alignment diagram for functional groups (left) that are specific to the MAIN IDEA model with their alignment to feature groups (right). Only functional groups and feature groups are shown if they have a positive correlation greater than 0.25 (blue edges) or a negative correlation less than  $-0.25$  (red edges). The numbers correspond to the IDs of the functional group or feature group that the node represents (see Table 3).

The types of feature groups that were aligned varied considerably between different comparisons.

### 5.2 Functional Component Groups

The initial extraction of functional components for each model elicited numbers of functional components between 28 and 119. Table 1 and 2 show that for a given model, fewer functional components will be extracted given a fewer instances in the analysis dataset. Despite this noise, a clear pattern emerges where the ORGANIZATION model has the most functional components, followed by the LANGUAGE model. The MAIN IDEA model has fewer functional components, with the SUPPORT model having the fewest.

When performing the dimensionality reduction to compute the functional groups, there is a consistent reduction to approximately 61-71% of the combined total functional components.

### 5.3 Important Functional Groups

Despite the variance in the number of feature groups and functional groups extracted per comparison, there is a remarkably consistent number of

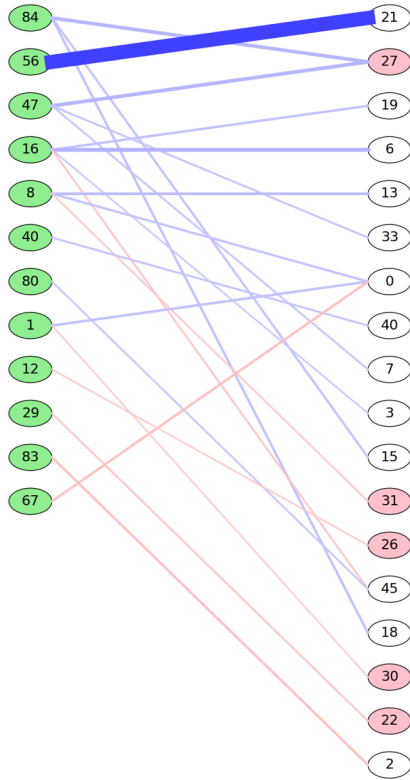


Figure 3: Alignment diagram for functional groups (left) that are common to both the LANGUAGE and MAIN IDEA models with their alignment to feature groups (right). Only functional groups and feature groups are shown if they have a positive correlation greater than 0.25 (blue edges) or a negative correlation less than  $-0.25$  (red edges). The numbers correspond to the IDs of the functional group or feature group that the node represents (see Table 3).

important functional groups that have at least one sufficient alignment to a feature group (Table 2). With the exception of the LANGUAGE v SUPPORT comparison, all other comparisons had between 21 and 29 aligned functional groups.

As a visual aid for the important functional groups, see the left sides of Figures 2 and 3. Each Figure is derived from the functional groups and feature groups of the LANGUAGE v MAIN IDEA comparison. The numbers on each node are the identifiers of a given functional group, a subset of which are represented in Table 3.

#### 5.4 Alignment of Functional Groups

The entirety of findings from the alignments for all of the comparisons would be too numerous to present in a conference paper format. However, we will present the major trends we found in our analysis. The first main trend is that all models had functional groups that we correlated with the

---

#### Functional Group 46

Diff:LANGUAGEVSMAINIDEA  $r = -0.39(p < 0.001)$

Independent Feature Group 1  $r = -0.43(p < 0.001)$

ModelErrors:MAINIDEA(+), ModelPairDifference(+), ModelErrors:LANGUAGE(-)

---

#### Functional Group 56

Predictions:MAINIDEA  $r = -0.13(p < 0.05)$

Independent Feature Group 21  $r = 0.75(p < 0.001)$

EssayStats:STDDEVSENTENCELENGTH(+), EssayStats:NUMSENTENCES(+), EssayStats:MEANWORDLENGTH(+), EssayStats:NUMWORDS(-), EssayStats:NUMPARAGRAPHS(-), EssayStats:MEANSENTENCELENGTH(-)

---

#### Functional Group 92

Predictions:LANGUAGE  $r = -0.13(p < 0.05)$

Independent Feature Group 12  $r = -0.20(p < 0.001)$

WordCluster:PRIORITIES(+), WordCluster:POPULATIONCOMPARISON(+), WordCluster:EFFICIENCY(+), WordCluster:TEENVALUES(-), WordCluster:STORYTELLING(-), WordCluster:SCHOOL(-), WordCluster:PARENTALDECISIONS(-), WordCluster:INFORMAL(-), WordCluster:HISTORICALCONFLICT(-)

---

Independent Feature Group 69  $r = 0.22(p < 0.001)$

RST:NNICONTRAST(+), RST:SNIEVALUATION(NSIELABORATION, LEAF)(+), RST:SNIBACKGROUND(LEAF, NSIELABORATION)(+), RST:NSIEVIDENCE(LEAF, NNICONJUNCTION)(+), RST:NNIJUNCTION(NNICONJUNCTION, NNIJUNCTION)(+), RST:NNICONTRAST(LEAF, LEAF)(+), RST:NNICONJUNCTION(NSIELABORATION, NNICONJUNCTION)(+), RST:SNIEVALUATION(NNICONJUNCTION, LEAF)(-), RST:NNICONJUNCTION(LEAF, LEAF)(-)

---

Table 3: Selected examples of correlated functional group/feature groups. Pearson’s R values for relevant importance metric (model difference, model predictions) and feature group alignment are presented with p-values.

statistical features of the essay. Furthermore, by computing the correlations between the individual features within that type, it was determined that *number of paragraphs* is likely the most salient contributor.

The second set of trends is presented in Table 4, where the percent of the total aligned feature groups per model was computed. This revealed that the ORGANIZATION model had considerably more aligned RST-based features than the other models, while the MAIN IDEA model had the least proportion. The LANGUAGE model had the most aligned word list features, which is the combination of the algorithmically and human-created word list features. For the last percentage, we combine the prompt and demographic features and find that the SUP-

Model	%RST	%Word	%Demo. &
		List	Prompt
ORGANIZATION	41	13	21
LANGUAGE	30	26	19
SUPPORT	36	19	13
MAIN IDEA	23	21	23

Table 4: % of aligned feature groups for a given model by feature type.

PORT model tended to align with fewer of these types of features. The reason for combining the demographic and prompt features is discussed in Section 6.

### 5.5 Qualitative Analysis

While the method that we presented can quickly advance one’s understanding of a model from the black-box neural network to aligned feature groups directly, understanding what function a feature group represents can be more difficult. It is thus necessary to resolve what a feature group represents to form a strong statement on what the model is doing. For instance, we found it concerning that so many of the models were connected with feature groups that contained demographic features (colored red in Figures 2 and 3). However, a qualitative look at the datasets for which prompts were included, we found that the distribution of prompts over the different schools, when controlling for essay length, were such that certain schools (with their demographic features) were the only source of certain prompts. It, therefore, becomes likely that many of these feature groups are more topic-based rather than the potentially more problematic demographic-based. This interpretation was reinforced by many of the feature groups with demographic information also including prompts (e.g. “*Independent Feature Group 29*” from Table 3) and by examining essays that present those feature groups.

## 6 Discussion

The results presented in the preceding section demonstrate the efficacy of the proposed method in extracting salient feature groups and functional groups from the neural models, particularly when applied to the dataset under consideration. The true potential of this method, however, lies in its capacity to be broadly applied to any neural AES system, thereby facilitating a deeper understanding of the models and the underlying processes they employ.

In the following discussion, we will delve further into the results, emphasizing the prominent trends observed in the alignment of functional groups and their correlation with essay features, as well as the implications of these findings for enhancing the interpretability and transparency of neural AES systems.

### 6.1 Functional Component and Feature Groups

The proposed method successfully extracted meaningful functional groups from the analyzed neural models. Notably, the LANGUAGE v SUPPORT comparison emerged as an outlier in several of our analyses. This discrepancy is likely attributable to the considerably fewer essays shared by both models’ analysis sets, which may result in a noisier analysis and expose a limitation of the method. As the size of the analysis increases, one would expect the extraction of feature groups and function groups to approach their ideal independence characteristics. Despite this limitation, the method managed to condense the analysis space from thousands of activations to fewer than 125 while still accounting for over 90% of the model’s variance.

Interestingly, the ORGANIZATION model exhibited the highest number of functional groups. This observation suggests that capturing the ORGANIZATION trait is a more intricate process, necessitating the learning of additional features. This notion is further corroborated by the comparisons between ORGANIZATION and other models; models which displayed very few, if any, functional groups exclusively present in the non-organization models.

### 6.2 Alignment of Important Functional Groups

In line with our expectations, the ORGANIZATION model demonstrated the greatest alignment with the RST tree features, while the LANGUAGE model displayed the most significant alignment with the word list features. It was postulated that ORGANIZATION would necessitate the model to possess knowledge of how ideas within essays are structured in relation to each other, a type of knowledge encoded by rhetorical structure theory. Although the RST parse trees recovered from the parser are considerably noisy (RST parsing of student essay data has been shown to be markedly more challenging than standard datasets (Fiacco et al., 2022)), the signal remained significant. Furthermore, we anticipated that the LANGUAGE model would have a

greater reliance on word choice, a concept mirrored by the word list-based feature groups.

Contrary to our expectations, the MAIN IDEA model exhibited the highest number of prompt-based feature groups. Our most plausible explanation for this observation is that certain prompts might have clearer expectations for thesis statements than others, a notion generally supported by a qualitative examination of the essays from prompts that score higher on MAIN IDEA.

## 7 Conclusion

The neural network interpretation technique presented in this paper demonstrates significant promise in learning the implicit rubrics of neural automated essay scoring models. By effectively mapping the intricate relationships between feature groups and the functional groups of the underlying scoring mechanism, the technique provides a step towards an understanding of the factors contributing to a transformer’s evaluation of essay quality. This enhanced understanding enables researchers and educators to not only identify potential biases in scoring models, but also to refine their models to ensure a more reliable and fair assessment of student performance.

The code for this method will be released and incorporated into an analysis tool for application to neural models not limited to the ones examined in this work with the goal to pave the way for the development of more transparency in neural AES models. These advancements can contribute to the overarching goal of promoting ethical and responsible AI in education by facilitating the examination and comprehension of complex neural models.

## Acknowledgements

This work was supported in part by NSF grant DRL 1949110.

## References

- Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *arXiv preprint arXiv:1608.04207*.
- Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.
- Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Peter F Brown, Vincent J Della Pietra, Peter V Desouza, Jennifer C Lai, and Robert L Mercer. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–480.

Steven Cao, Victor Sanh, and Alexander M Rush. 2021. Low-complexity probing via finding subnetworks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 960–966.

Pierre Comon. 1994. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single  $\&!#\ast$  vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136.

Leon Derczynski, Sean Chester, and Kenneth S Bøgh. 2015. Tune your brown clustering, please. In *International Conference Recent Advances in Natural Language Processing, RANLP*, volume 2015, pages 110–117. Association for Computational Linguistics.

Yuning Ding, Brian Riordan, Andrea Horbach, Aoife Cahill, and Torsten Zesch. 2020. Don’t take “nswvt-nvakgxpm” for an answer—the surprising vulnerability of automatic content scoring systems to adversarial input. In *Proceedings of the 28th international conference on computational linguistics*, pages 882–892.

James Fiacco, Samridhi Choudhary, and Carolyn Rose. 2019. Deep neural model inspection and comparison via functional neuron pathways. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 5754–5764.

James Fiacco, Shiyang Jiang, David Adamson, and Carolyn Rose. 2022. Toward automatic discourse parsing of student writing motivated by neural interpretation. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 204–215.

Harold Hotelling. 1933. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417.

Shi Huawei and Vahid Aryadoust. 2023. A systematic review of automated writing evaluation systems. *Education and Information Technologies*, 28(1):771–795.

Shiyang Jiang, Kexin Yang, Chandrakumari Suvarna, Pooja Casula, Mingtong Zhang, and Carolyn Rose. 2019. Applying rhetorical structure theory to student essays for providing automated writing feedback. In *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019*, pages 163–168.



- Cancan Jin, Ben He, Kai Hui, and Le Sun. 2018. Tdnn: a two-stage deep neural network for prompt-independent automated essay scoring. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1088–1097.
- Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. 2018. Sharp nearby, fuzzy far away: How neural language models use context. *arXiv preprint arXiv:1805.04623*.
- Yaman Kumar, Mehar Bhatia, Anubha Kabra, Jessy Junyi Li, Di Jin, and Rajiv Ratn Shah. 2020. Calling out bluff: attacking the robustness of automatic scoring systems with simple adversarial testing. *arXiv preprint arXiv:2007.06796*.
- Yaman Kumar, Swapnil Parekh, Somesh Singh, Junyi Jessy Li, Rajiv Ratn Shah, and Changyou Chen. 2023. Automatic essay scoring systems are both overstable and oversensitive: Explaining why and proposing defenses. *Dialogue & Discourse*, 14(1):1–33.
- Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. 2018. Lstms can learn syntax-sensitive dependencies well, but modeling structure makes them better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1426–1436.
- Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.
- William C Mann and Sandra A Thompson. 1987. *Rhetorical structure theory: A theory of text organization*. University of Southern California, Information Sciences Institute Los Angeles.
- Atsushi Mizumoto and Masaki Eguchi. 2023. Exploring the potential of using an ai language model for automated essay scoring. *Research Methods in Applied Linguistics*, 2(2):100050.
- Farah Nadeem, Huy Nguyen, Yang Liu, and Mari Ostendorf. 2019. Automated essay scoring with discourse-aware neural models. In *Proceedings of the fourteenth workshop on innovative use of NLP for building educational applications*, pages 484–493.
- Ellis B Page. 1966. The imminence of... grading essays by computer. *The Phi Delta Kappan*, 47(5):238–243.
- Dadi Ramesh and Suresh Kumar Sanampudi. 2022. An automated essay scoring systems: a systematic literature review. *Artificial Intelligence Review*, 55(3):2495–2527.
- 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.
- Maria Ryskina and Kevin Knight. 2021. Learning mathematical properties of integers. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 389–395.
- Angad Sethi and Kavinder Singh. 2022. Natural language processing based automated essay scoring with parameter-efficient transformer approach. In *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*, pages 749–756. IEEE.
- Xing Shi, Kevin Knight, and Deniz Yuret. 2016. Why neural translations are the right length. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2278–2282.
- Kaveh Taghipour and Hwee Tou Ng. 2016. A neural approach to automated essay scoring. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1882–1891.
- Patti West-Smith, Stephanie Butler, and Elijah Mayfield. 2018. Trustworthy automated essay scoring without explicit construct validity. In *AAAI Spring Symposia*.
- Bronwyn Woods, David Adamson, Shayne Miel, and Elijah Mayfield. 2017. Formative essay feedback using predictive scoring models. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 2071–2080.
- Xunjie Zhu, Tingfeng Li, and Gerard Melo. 2018. Exploring semantic properties of sentence embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 632–637.

# Analyzing Bias in Large Language Model Solutions for Assisted Writing Feedback Tools: Lessons from the Feedback Prize Competition Series

Perpetual Baffour, Tor Saxberg and Scott Crossley

## Abstract

This paper analyzes winning solutions from the Feedback Prize competition series hosted from 2021-2022. The competitions sought to improve Assisted Writing Feedback Tools (AWFTs) by crowdsourcing Large Language Model (LLM) solutions for evaluating student writing. The winning LLM-based solutions are freely available for incorporation into educational applications, but the models need to be assessed for performance and other factors. This study reports the performance accuracy of Feedback Prize-winning models based on demographic factors such as student race/ethnicity, economic disadvantage, and English Language Learner status. Two competitions are analyzed. The first, which focused on identifying discourse elements, demonstrated minimal bias based on students' demographic factors. However, the second competition, which aimed to predict discourse effectiveness, exhibited moderate bias.

## 1 Introduction

Assisted writing feedback tools (AWFTs) are a promising example of educational applications using Natural Language Processing (NLP) algorithms that can innovate and accelerate student learning (Nunes, Cordeiro, Limpo, & Castro, 2022). Recent advances in large language models (LLMs) have increased AWFTs' capabilities to process and provide feedback on student writing with human-like sophistication (Kasneji et al., 2023). The Feedback Prize competition series, hosted on Kaggle in 2021-2022, was an important step in advancing AWFTs potential by crowdsourcing innovative LLM solutions for

assessing and evaluating student writing that were open science (The Learning Agency Lab, n.d.).

The competitions were a success with over 6,000 teams participating and over 100,000 open-source algorithms developed. (The Learning Agency Lab, n.d.) However, these algorithms have not been reported outside of the Kaggle interface, limiting knowledge of their use and minimizing potential adoption into educational applications. Additionally, the algorithms have not been assessed for bias, which may limit their effectiveness in a classroom setting, especially if that bias is aimed towards student populations that have been historically marginalized. The purpose of this study is to report initial performance for the winning Feedback Prize models and to disaggregate performance accuracy in demographic factors including race/ethnicity, economic disadvantage, and English Language Learner (ELL) status.

## 2 PERSUADE Corpus

The first two competitions in the Feedback Prize series were based on the PERSUADE (Persuasive Essays for Rating, Selecting, Analyzing, and Understanding Discourse Elements) corpus, a collection of ~25,000 argumentative essays written by students in the U.S. in grades 6 through 12 (Crossley et al., 2022). The essays were annotated by experts for discourse elements and the effectiveness of the discourse elements. Discourse elements refer to a span of text that performs a specific rhetorical or argumentative function, while discourse effectiveness is a rating of the quality of the discourse element in supporting the writer's overall argument. The effectiveness scale included Ineffective, Adequate, and Effective ratings. The annotation scheme for discourse elements is based on an adapted or simplified version of the Toulmin argumentative framework (Stapleton & Wu, 2015).

The discourse elements that were annotated for each essay were:

- **Lead.** An introduction begins with a statistic, a quotation, a description, or some other device to grab the reader’s attention and point toward the thesis.
- **Position.** An opinion or conclusion on the main question.
- **Claim.** A claim that supports the position.
- **Counterclaim.** A claim that refutes another claim or gives an opposing reason to the position.
- **Rebuttal.** A claim that refutes a counterclaim.
- **Evidence.** Ideas or examples that support claims, counterclaims, rebuttals, or the position.
- **Concluding Statement.** A concluding statement that restates the position and claims.

The essays were annotated using a rigorous, double-blind rating process with 100 percent adjudication, such that each essay was independently reviewed by two expert raters and adjudicated by a third rater. Overall inter-rater agreement for discourse elements assessed using a weighted Cohen’s Kappa was 0.73, which indicates relatively high reliability. While the experts who annotated the corpus for discourse elements also rated each element’s effectiveness in supporting the writer’s argument, misalignment in segmentation between the raters in the discourse elements make it difficult to calculate inter-rater reliability for the effectiveness labels.

### 3 Feedback Prize 1.0 Models

The first Feedback Prize competition, (Feedback Prize 1.0: Evaluating Student Writing) was hosted on Kaggle and involved the tasks of segmenting essays into smaller sections and assigning each section a discourse label such as lead, position, claim, and evidence. To evaluate performance, submissions were assessed based on the word overlap between ground truth and predicted outputs. A model prediction was considered correct (true positive) if there was at least a 50% word overlap between the machine-segmented section and the human-segmented section, as well as a match between their discourse label. False negatives were unmatched ground truths, and false positives were unmatched predictions. The final score was calculated by

Table 1: True positive rate (TPR) by English Language Learner status of student writer, Feedback Prize 1.0 2<sup>nd</sup> place

Status	N	TPR	SD
ELL	7,565	0.717	0.235
Not ELL	81,207	0.726	0.220
All	88,772	0.725	0.221

Table 2: True positive rate (TPR) by economic status of student writer, Feedback Prize 1.0 2<sup>nd</sup> place

Status	N	TPR	SD
Disadvantaged	35,696	0.713	0.226
NDA	42,698	0.743	0.214
All	78,394	0.729	0.221

\*Note: NDA refers to non-disadvantaged students.

determining the number of true positives, false positives, and false negatives for each class (i.e., discourse label) and taking the macro F1 score across all classes.

The analysis in this paper examines the second-place, third-place, and sixth-place winning solutions from this competition. Overall, the winning solutions were broadly based on ensembles of large-scale, pre-trained Transformers, paired with custom pre-processing and post-processing techniques to improve accuracy. The first-place model was not analyzed because its complexity made it difficult to replicate and impractical in educational settings. The overall macro F1 score did not differ significantly between the second-place, third-place, and sixth-place solutions, with values of .740, .740, and .732, respectively.

To assess potential bias in the models, performance accuracy was further disaggregated by demographic factors (race/ethnicity, English Language Learner status, and economic disadvantage) and discourse effectiveness (Ineffective, Adequate, Effective). Specifically, T-tests and ANOVAs indicated that the average true positive rate (TPR) per essay of the second-place, third-place, and sixth-place models significantly varied based on demographic factors, but the effect sizes were small (see Tables 1-3). None of the t-tests or ANOVA tests reported any results with a p-value < 0.01 and a Cohen’s d > 0.2. For instance, the t-test comparing TPR differences between ELL and non-ELL writing showed a p-value of 0.03 and Cohen’s d of 0.103 for the second-place model,

Table 3: True positive rate (TPR) by race/ethnicity of student writer, Feedback Prize 1.0 2<sup>nd</sup> place

Race/Ethnicity	N	TPR	SD
White	42,197	0.723	0.217
Black	17,060	0.722	0.228
Hispanic	23,055	0.712	0.229
Asian	6,814	0.777	0.198
American Indian	574	0.728	0.226
Multiple	3,884	0.743	0.197
All	93,584	0.726	0.221

suggesting a negligible difference in model performance.

#### 4 Feedback Prize 2.0 Models

The second Feedback Prize competition (Feedback Prize 2.0: Predicting Effective Arguments) also hosted on Kaggle required models to predict the effectiveness rating of discourse labels, using multi-class logarithmic loss as the evaluation metric. More specifically, for each discourse label, the model had to submit the probabilities (or the likelihood) that the label belongs to each of the three effectiveness ratings (Ineffective, Adequate, Effective). The closer the predicted probabilities were to the actual true label, the higher the model score would be. Feedback Prize 2.0 also prioritized computationally efficient algorithms, with a prize-incentivized “Efficiency Track” that evaluated submissions for both accuracy and speed.

Feedback Prize 2.0 comprised a smaller subset of the data from the first competition (around 6,900 out of the 26,000 essays), due to a need for greater balance in effectiveness scores. In the complete PERSUADE corpus, only 4% of discourse elements were labeled Ineffective while 80% were labeled Adequate and 16% were labeled Effective. The subset used in Feedback Prize 2.0 corpus had a distribution of 18% Ineffective, 24% Effective, and 58% Adequate, resulting in greater balance.

The analysis presented in this paper examines the performance of the winning models (first, second, and third place) in the Efficiency Track on the competition test set. A common trend among winning solutions from the Efficiency Track was to fine-tune a single pre-trained Transformer model on the competition dataset to minimize space and runtime requirements. The authors did not analyze the winners from the non-efficiency track because performance was similar, but computational demands were much higher. The

Figure 1: Performance accuracy by ELL status of student writer and discourse effectiveness label, Feedback Prize 2.0 Efficiency Track 1<sup>st</sup> place

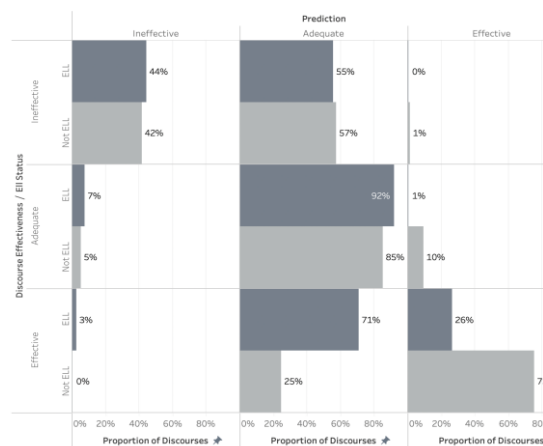
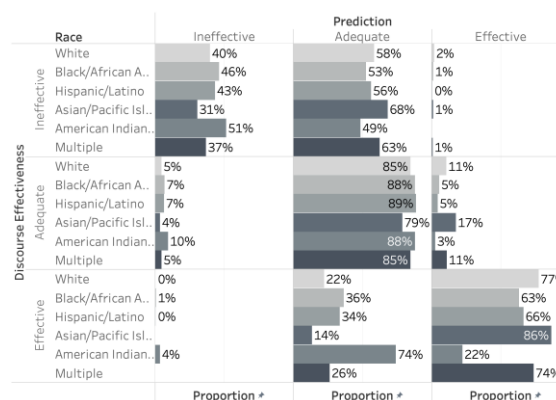


Figure 2: Performance accuracy by race/ethnicity of student writer and discourse effectiveness label, Feedback Prize 2.0 Efficiency Track 1<sup>st</sup> place



analysis consists of two parts. The first part examines the accuracy of the models in predicting the three original effectiveness ratings (Ineffective, Adequate, Effective). In the second part, the winning models' predictions were evaluated by grouping Ineffective and Adequate labels into a Non-Effective label, creating a binary outcome variable (Effective, Non-Effective). This analysis recoded the labels 'post hoc,' after the model submitted probabilities for all three original ratings. In both analyses, the model's predicted label was determined as the label with the highest predicted likelihood among the outputted probabilities.

#### 4.1 Analysis of accuracy using original effectiveness ratings

The first part of the Feedback Prize 2.0 bias analysis found that the selected winning models

showed higher levels of bias for certain students compared to the winning models from Feedback Prize 1.0. This disparity can be attributed to patterns in the label distribution of the data. The data sample for the Feedback Prize 2.0 competition had a more balanced representation of minority and historically disadvantaged students in the overall sample, but there were roughly twice as many discourse elements labeled Ineffective from economically disadvantaged students and almost three times as many Effective discourses from non-disadvantaged students.

As a result, effective writing discourses from white, non-ELL, and economically advantaged students were more likely to receive higher ratings and the models amplified the existing disproportionate representation of effective writing found in the human-rated dataset. As shown in Figure 1, the first-place model was more accurate in identifying effective discourses in non-ELL writing (76% vs 27% accurate) with a statistically significant difference in likelihood scores (p-value  $\sim 0.000$ ) and a larger effect size (Cohen's  $d \sim 0.671$ ), as shown in Table 4. As shown in Table 5, the first-place model was also less accurate in predicting effective writing for economically disadvantaged students, and a t-test revealed that the difference in likelihood scores for effective discourses was statistically significant (p-value  $\sim 0.000$ ) and the effect size was moderate (Cohen's  $d \sim 0.263$ ). Similarly, accuracy disaggregated by the race/ethnicity of each student writer also showed statistically significant differences (p-values  $\sim 0.000$ ), but with small effect sizes (Cohen's  $d \sim 0.15$ ), as shown in Table 6 and Figure 2.

Table 4: Likelihood scores for effective discourses by English Language Learner status of student writer, Feedback Prize 2.0 Efficiency Track 1<sup>st</sup> place

Status	N	Likelihood	SD
ELL	2,623	0.028	0.083
Not ELL	19,853	0.246	0.321
All	22,476	0.221	0.311

Table 5: Likelihood scores for effective discourses by economic status of student writer, Feedback Prize 2.0 Efficiency Track 1<sup>st</sup> place

Status	N	Likelihood	SD
Disadvantaged	10,268	0.113	0.224
NDA	9,805	0.338	0.353
All	20,073	0.223	0.315

\*Note: NDA refers to non-disadvantaged students.

Table 6: Likelihood scores for effective discourses by race/ethnicity of student writer, Feedback Prize 2.0 Efficiency Track 1<sup>st</sup> place

Race/ethnicity	N	Likelihood	SD
White	9,816	0.270	0.328
Black	4,157	0.133	0.246
Hispanic	6,218	0.149	0.261
Asian	1,721	0.398	0.370
Am. Ind.	179	0.096	0.176
Multiple	888	0.250	0.321
All	22,979	0.220	0.310

#### 4.2 Analysis of accuracy using binary label of effectiveness

The second part of the analysis aimed to address the low sample size of Ineffective discourses in the dataset by recoding the effectiveness label as a binary variable. This involved combining Ineffective and Adequate discourses into a Non-Effective label. The goal was to examine whether similar levels of bias persisted in the recoded label. Combining Adequate and Ineffective discourse labels into a Non-Effective category did achieve greater balance in performance accuracy for the Non-Effective label, but there remained bias in the prediction of Effective discourses because white, non-ELL, and advantaged students remain overrepresented in this category, as shown in Figure 3.

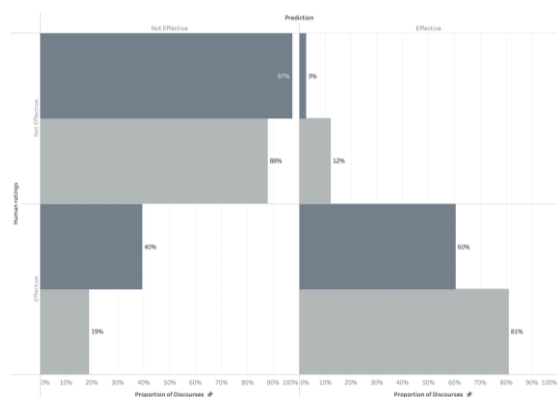
### 5. Discussion

The winning solutions across the first two Feedback Prize competitions reported a degree of accuracy comparable to that of humans, which is an important indicator of the models' strength. Additionally, since the models are open-source, they can quickly be adapted into educational applications to not only assess student writing at a summative level but to also provide fine-grained feedback to students at the formative level.

However, as noted in the analyses above, the winning solutions from the second competition that focused on predicting effective arguments showed a moderate degree of bias among factors related to race/ethnicity, economic status, and English Language Learner (ELL) status while the winning solutions from the first competition, which focused on annotating discourse elements, showed minimal bias.

It appears the models from Feedback Prize 2.0 amplified the biases inherent in the data despite not being explicitly trained with demographic

Figure 3: Performance accuracy for Non-Effective and Effective discourses, by student economic status, Feedback Prize 2.0 Efficiency Track 1<sup>st</sup> place



information. Data bias in label distribution, label agreement, and demographic representation in the PERSUADE corpus may have contributed to the model bias, but it is unclear how well these factors could be addressed given current writing achievement disparities in the U.S. educational system (National Center for Education Statistics, 2012). Using a binary classification for effectiveness (i.e., recoding the data as Effective or Ineffective) helped to mitigate the bias in the models to some degree. However, the use of models from Feedback Prize 2.0 for educational applications should be handled with care, especially when dealing with students from diverse populations.

These analyses demonstrate the importance of assessing algorithms for bias prior to wide-scale adoption. The results point to future work in building educational NLP applications like AWFTs to identify potential data biases in label distribution, agreement, or demographic representation before adoption to reduce bias in algorithmic outputs and help ensure fairness in systems. As can be seen with the PERSUADE corpus, bias will likely be present in any dataset that accurately represents populations in the United States because of achievement disparities in the educational systems.

## References

Scott A. Crossley, Perpetual Baffour, Yu Tian, Aigner Picou, Meg Benner, and Ulrich Boser. 2022. The persuasive essays for rating, selecting, and understanding argumentative and discourse

elements corpus 1.0. *Assessing Writing*, 54. <https://doi.org/10.1016/j.asw.2022.100667>

Kevin A. Hallgren. 2012. Computing inter-rater reliability for observational data: an overview and tutorial. *Tutorials in Quantitative Methods for Psychology*, 8(1):23–34. <https://doi.org/10.20982/2Ftqmp.08.1.p023>

Enkelejda Kasneci, Kathrin Sessler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Gunnemann, Eyke Hüllermeier, Stepha Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, and Tina Seidel. 2023. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103. <https://doi.org/10.1016/j.lindif.2023.102274>

The Learning Agency Lab. (n.d.). The Feedback Prize: A case study in assisted writing feedback tools working paper. <https://www.the-learning-agency-lab.com/the-feedback-prize-case-study/>

National Center for Education Statistics. (2012). The Nation's Report Card: Writing 2011 (NCES 2012-470). National Center for Education Statistics.

Andreia Nunes, Carolina Cordeiro, Teresa Limpo, and São Luís Castro. 2021. Effectiveness of automated writing evaluation systems in school settings: A systematic review of studies from 2000 to 2020. *Journal of Computer Assisted Learning*, 38(2):599–620. <https://doi.org/10.1111/jcal.12635>

E. Michael Nussbaum, CarolAnne M. Kardash, and Steve Graham. 2005. The effects of goal instructions and text on the generation of counterarguments during writing. *Journal of Educational Psychology*, 97(2):157–169. <https://psycnet.apa.org/doi/10.1037/0022-0663.97.2.157>

Paul Stapleton and Yanming (Amy) Wu. 2015. Assessing the quality of arguments in students' persuasive writing: A case study analyzing the relationship between surface structure and substance. *Journal of English for Academic Purposes*, 17:12–23. <https://doi.org/10.1016/j.jeap.2014.11.006>

# Improving Reading Comprehension Question Generation with Data Augmentation and Overgenerate-and-rank

Nischal Ashok Kumar<sup>1</sup>, Nigel Fernandez<sup>1</sup>, Zichao Wang<sup>2</sup>, Andrew Lan<sup>1</sup>

University of Massachusetts Amherst<sup>1</sup>, Adobe Research<sup>2</sup>

{nashokkumar, nigel, andrewlan}@cs.umass.edu, jackwa@adobe.com

## Abstract

Reading comprehension is a crucial skill in many aspects of education, including language learning, cognitive development, and fostering early literacy skills in children. Automated answer-aware reading comprehension question generation has significant potential to scale up learner support in educational activities. One key technical challenge in this setting is that there can be multiple questions, sometimes very different from each other, with the same answer; a trained question generation method may not necessarily know which question human educators would prefer. To address this challenge, we propose 1) a data augmentation method that enriches the training dataset with diverse questions given the same context and answer and 2) an overgenerate-and-rank method to select the best question from a pool of candidates. We evaluate our method on the FairytaleQA dataset, showing a 5% absolute improvement in ROUGE-L over the best existing method. We also demonstrate the effectiveness of our method in generating harder, “implicit” questions, where the answers are not contained in the context as text spans.

## 1 Introduction

Reading comprehension is crucial in assessing students’ language learning ability and complex reasoning skills. Comprehending and interpreting stories such as fairy tales, with specific emphasis on narratives, foster early intellectual and literacy development in children (Sim and Berthelsen, 2014; Lynch et al., 2008). Asking suitable educational-focused questions can help students understand the context of the fairy tales better and inspire their interests (Ganotice Jr et al., 2017; Zevenbergen and Whitehurst, 2003; Xu et al., 2021). However, constructing suitable questions at scale is hard since it is both time intensive and cognitively challenging (Golinkoff et al., 2019). Researchers have developed models that can automatically generate

questions or question-answer pairs to meet the demand for a large pool of relevant questions (Kurdi et al., 2020; Yao et al., 2022). These advances can potentially facilitate the development of artificial intelligence (AI)-supported learning platforms to help students develop reading comprehension skills (Zhang et al., 2022).

Prior work on question generation in educational applications can be broadly classified into two categories: *answer-aware*, which is the focus of our current work, and *answer-unaware* (see Dugan et al. (2022) for a feasibility study), depending on whether the desired answer is given or not. For answer-aware question generation, the goal is to build an AI-based system to generate a question given both the context and the answer (Wang et al., 2018). The context can be any text segment, from a few sentences to a possibly long document, that provides background information on which the question is grounded in. The answer is a short span of text that is either part of the context (explicit) or not part of the context but can be inferred from the context (implicit). More specifically, in answer-aware question generation, the question generation system is trained using the context-answer pairs as input and the question as the output (Yao et al., 2022). See Section 2 for a detailed discussion on related work.

A key challenge in answer-aware question generation is that there are often multiple relevant questions for a given context-answer pair. Existing question generation systems are limited in identifying which questions human educators would prefer from multiple relevant ones. Table 1 shows an example context-answer pair from the FairytaleQA dataset (Xu et al., 2022b) with four relevant questions that can be answered by “a lovely dinner”, the given answer. The first and second questions focus on describing the setting of the context framed using the object (table) and the subject (Tom and Hunca), respectively. The third question adds a

causal element inquiring about the cause of Tom and Hunca’s emotion. The fourth question is predictive in nature, asking about an event which can be inferred from the context.

Selecting the top question from multiple relevant and diverse question candidates is challenging. For a question generation system to perform this challenging task, it needs to be able to both generate diverse and valid question candidates and also accurately rank and select the top question. To generate diverse question candidates, a question generation system needs to be trained on multiple different relevant questions for a given context-answer pair. To accurately select the top question, a question generation system needs to learn to rank the question candidates by matching the preferences of human educators. We incorporate both of these ideas in our proposed methods in this work.

### 1.1 Contributions

In this paper, we detail two novel methods to improve the robustness of automated answer-aware reading comprehension question generation. We validate their effectiveness through both quantitative and qualitative experiments on the FairytaleQA dataset (Xu et al., 2022b); we make our implementation publicly available.<sup>1</sup> Built on top of a Flan-T5 (Chung et al., 2022) fine-tuning backbone, our contributions are summarized as follows:

- We propose a **data augmentation** method to augment the training set with synthetically generated diverse and relevant questions. Specifically, we prompt a larger language model, OpenAI Codex (Chen et al., 2021), to first generate a diverse question pool and then filter out questions that are inconsistent with the given context-answer pair using a question-answering model.
- We propose an **overgenerate-and-rank** method to rank multiple generated question candidates for the given context-answer pair. Specifically, we fine-tune a separate BERT-based model by optimizing a distribution matching objective to learn which questions are more preferable to human educators and use the model to rank them.

<sup>1</sup>The code for the paper can be found at: <https://github.com/umass-ml4ed/question-gen-aug-ranking>

Context	Tom Thumb and Hunca Munca went up-stairs and peeped into the dining room. Then they squeaked with joy. Such a lovely dinner was laid out upon the table ...
Answer	a lovely dinner
Questions	<ol style="list-style-type: none"> <li>1. What was laid upon the table?</li> <li>2. What did Tom and Hunca see in the dining room?</li> <li>3. What made Tom and Hunca squeak with joy?</li> <li>4. What will Tom and Hunca enjoy eating in the dining room?</li> </ol>

Table 1: Example context-answer pair from the FairytaleQA dataset with multiple valid questions.

- We conduct extensive experiments to validate the effectiveness of our methods. Our best method achieves a 5% absolute increase in the ROUGE-L score over the best existing baseline (Xu et al., 2022b). We also observe that 1) the data augmentation method can be used to balance questions of different types in the training data and 2) the overgenerate-and-rank method is particularly effective at generating harder questions, i.e., those with answers not explicitly present in the context as text spans.

## 2 Related Work

### 2.1 QA Datasets on Narratives

There have been several works proposing QA and QG datasets of educational importance. NarrativeQA (Kočiskỳ et al., 2018) requires students to answer questions written by crowd workers based on books or movie scripts. TellMeWhy (Lal et al., 2021) is another dataset that contains only “why” based questions that need additional information not directly present in the text to be answered. A recent and popular dataset to facilitate assessment and training of students’ narrative comprehension skills is the FairytaleQA (Xu et al., 2022b) dataset. FairytaleQA contains question-answer pairs written by education experts on fairy tale stories obtained from Project Gutenberg<sup>2</sup>. FairytaleQA is composed of questions focusing on several narrative elements. We validate the effectiveness of our question generation methods with extensive experiments on FairytaleQA.

<sup>2</sup><https://www.gutenberg.org>



## 2.2 Question Generation

There are several works on question generation for reading comprehension. Stasaski et al. (2021) and Zou et al. (2022) propose question generation methods based on causal relations and unsupervised learning, respectively. However, their methods are focused on very specific questions and are thus not generalizable. In contrast, our work focuses on a broad variety of questions covering different narrative elements in reading comprehension. Rathod et al. (2022) proposes to generate multiple semantically similar but lexically diverse questions for a given answer. However, their work is limited to generating only two questions per answer. In contrast, our approach is capable of generating multiple diverse and relevant questions, along with a ranking method to select the best question aligned with human educator preferences. Recent work on the FairytaleQA dataset develops event-based question generation methods (Zhao et al., 2022; Xu et al., 2022a). However, their results are reported on only a small subset of attributes: action, causal relationship, and outcome resolution. In contrast, we report our results over all attributes on the complete FairytaleQA dataset and compare with the current state-of-the-art baseline. Yuan et al. (2022) propose a prompt-based question generation method that leverages large language models (LM) like GPT-3. However, these black-box LMs have limited API only access. In contrast, our method uses open-source language models to achieve competitive results. The FairytaleQA dataset paper (Xu et al., 2022b) proposes the current state-of-the-art question generation method by fine-tuning the BART (Lewis et al., 2020) LM to generate the ground truth question given the input context-answer pair. Improving upon LM fine-tuning, we propose two question generation methods for increased robustness, data augmentation and overgenerate-and-rank, which are able to both generate diverse and valid question candidates and also accurately rank and select the top question aligned with human educator preference.

## 3 Methodology

In this section, we first introduce the problem setup for question generation on FairytaleQA (Xu et al., 2022b). We then detail our question generation approach, building upon the baseline of fine-tuning a language model, by adding our data augmentation method to augment the training set with diverse

questions, followed by our over-generate-and-rank method to select the top question from the diverse question candidates generated.

### 3.1 Problem Formulation and Dataset Details

FairytaleQA (Xu et al., 2022b) is a popular dataset for both question answering and question generation in the education community supporting narrative comprehension, targeting students from kindergarten to eighth grade. Written by education experts, FairytaleQA contains 10,580 question-answer pairs  $(q_i, a_i)$ , indexed by  $i$ , from 278 classical fairytale stories. Each question-answer pair is sourced from a section of a story referred to as the context  $c_i$ . The goal for a trained question generation model is to generate the ground truth question  $q_i$  conditioned on the input context-answer pair  $(c_i, a_i)$ .

Question-answer pairs in FairytaleQA can be categorized in two major ways: 1) by attributes and 2) by the source of answers. In attribute categorization, question-answer pairs capture seven different narrative elements or relations, referred to as attributes, which are character, setting, action, feeling, causal relationship, outcome resolution, and prediction. Orthogonal to the previous categories, questions can also be categorized by whether the answer span is explicitly contained within the context or is implicit and need to be inferred from the context. Explicit questions capture specific story facts while implicit questions require summarization and inference skills. FairytaleQA is imbalanced with respect to question attributes, with action and causal relationship questions accounting for 60% of the dataset. Our data augmentation method helps balance questions of different attributes.

### 3.2 Language Model Fine-tuning

We first describe our LM fine-tuning approach for question generation. We use a pre-trained Flan-T5 (Chung et al., 2022) model as our base LM for question generation. We also tried using vanilla T5 (Raffel et al., 2020) and GPT-2 (Radford et al., 2019) as our base LM which gave a comparable but lower performance, possibly because Flan-T5 is instruction fine-tuned on a large number of tasks relevant to both QA and QG. Therefore, for simplicity of exposition, we detail our question generation methods using Flan-T5 as the base LM. We construct the input using a combination of the context  $c_i$  and answer  $a_i$  with the following

template:      Generate question given  
context and answer:    Context:     $c_i$   
Answer:  $a_i$ .

Let  $\theta$  represent the LM parameters to be learned. We fine-tune our LM over all context-answer pairs  $(c_i, a_i)$  to generate the corresponding ground truth question  $q_i$  using a language modeling objective. The language modeling objective is the negative log-likelihood of generating the ground truth question calculated at the token level. The objective  $\mathcal{L}_i(\theta)$  for the  $i^{\text{th}}$  training sample is given by:

$$\mathcal{L}_i(\theta) = - \sum_t \log P(q_{i,t} | c_i, a_i, q_{i,<t}) \quad (1)$$

where  $q_{i,t}$  is the  $t^{\text{th}}$  token of question  $q_i$  and  $q_{i,<t}$  refers to all tokens preceding the  $t^{\text{th}}$  token. Our finetuning objective is the sum of this loss across all training questions.

### 3.3 Data Augmentation

For a question generation system to be robust in selecting the best question for context-answer pairs with multiple relevant questions, it must first be able to generate diverse and suitable question candidates for a given context-answer pair. Moreover, education experts who created the FairytaleQA dataset followed the pattern of first reading the context, then writing a question, and finally writing the answer. This process implies that there could often be multiple valid questions associated with the same context-answer pair in addition to the ground-truth question, which can be used to augment the dataset (as seen in Table 1). Therefore, we propose an automated data augmentation method to enrich the training set with diverse and relevant questions for each context-answer-question triplet. We prompt a larger LM, OpenAI Codex (Chen et al., 2021), in an in-context prompting fashion (Brown et al., 2020) to first generate diverse questions for each context-answer pair and then filter out unsuitable questions with consistency matching; we detail both steps below.

**Synthetic Data Generation.** We first generate synthetic data, i.e.,  $M = 4$  diverse question candidates  $\{\hat{q}_{i,1}, \dots, \hat{q}_{i,M}\}$  for each context-answer-question triplet  $(c_i, a_i, q_i)$  using the OpenAI Codex LM (Chen et al., 2021) in an in-context prompting fashion. We construct the in-context prompt by randomly selecting five context-answer-question triplets from the training set with the same attribute

as the target context-answer-question triplet to augment. We then append the target triplet followed by the prompt: “Another question with the same answer is”. These examples help Codex to adapt to the style of questions written by education experts. We use nucleus sampling (Holtzman et al., 2020) to generate synthetic questions with a p value of 0.9 and temperature of 0.8 to ensure diversity.

**Consistency Matching.** Since there is no guarantee that the generated questions are faithful and match the context-answer pair, we filter out *inconsistent* questions using a consistency matching criterion inspired from Wang et al. (2021). A generated question is consistent with respect to its input context-answer-question triplet if the answer of the generated question is the same as (or similar to) the input ground-truth answer. This consistency criterion enables us to include diverse yet consistent synthetic questions to augment the ground-truth questions during training.

To obtain the answer of a generated question, we again use Codex in an in-context prompting fashion with a subtle change in the prompt. We use the same five in-context examples of context-answer-question triplets taken from the same attribute as the target context-answer-question being augmented. However, we change the earlier context-answer-question pattern suitable for question generation and reformulate in the order of context-question-answer appropriate for question answering. We denote the answer to the generated question  $\hat{q}_{i,j}$  as  $\hat{a}_{i,j}$ . We use greedy decoding since we need the single best answer. We observe that comparing the similarity of this obtained answer generated by Codex to the ground truth answer  $a_i$  written by human education experts can sometimes exclude consistent synthetic questions incorrectly. We alleviate this issue by obtaining another reference answer to compare with; we prompt Codex in an in-context fashion to obtain the answer to the ground truth question  $q_i$ , which we denote as  $\bar{a}_i$ . Note that  $\bar{a}_i$  could be different from the ground truth answer  $a_i$  as shown in an example in Table 6 in the Supplementary Material.

To check consistency, we measure the similarity between  $\hat{a}_{i,j}$  and both  $a_i$  and  $\bar{a}_i$  using the ROUGE-1 F1 score (Lin, 2004). If either similarity is greater than a threshold of 0.5, we include the context-answer-synthetic question triplet  $(c_i, a_i, \hat{q}_{i,j})$  in our augmented training set. We outline our method in Figure 1 and also in Algorithm 1 in the Supplemen-

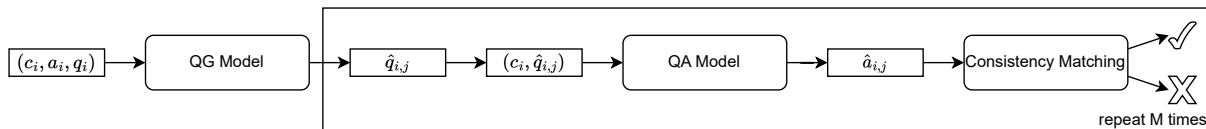


Figure 1: Our automated data augmentation method to enrich the training set with diverse and relevant questions for each context-answer-question triplet.

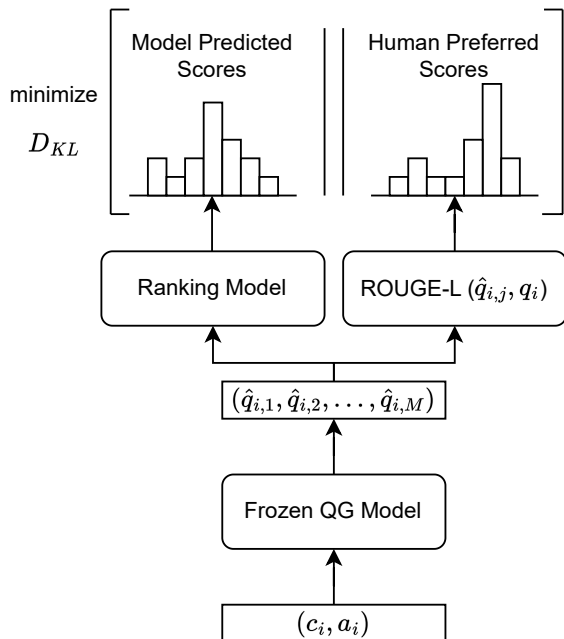


Figure 2: The training process of the ranking model used in our overgenerate-and-rank method with distribution matching-based ranking.

tary Material.

### 3.4 Overgenerate-and-Rank

Selecting the top question preferable to human educators from multiple relevant and diverse question candidates for the given context-answer pair is hard. We propose an overgenerate-and-rank method which first overgenerates several question candidates for each context-answer pair using the fine-tuned model (as described in Section 3.2). We use various decoding strategies, including nucleus sampling (Holtzman et al., 2020) and contrastive search (Su et al., 2022) to ensure diversity. We then rank these generated questions based on a criterion. We use two kinds of ranking methods, perplexity-based ranking and distribution matching-based ranking, which we detail below.

**Perplexity-based Ranking.** In this ranking method, we use perplexity as a metric to rank the generated questions. The perplexity of a language model given a question measures the uncertainty of

generating the question under that language model. The lower the perplexity of a question, the more probable is the question according to the language model. We first overgenerate  $K = 10$  questions for the given context-answer pair using nucleus sampling or contrastive search. We then compute the perplexity of these questions given the fine-tuned language model. We then select the question with the lowest perplexity as the best question for the given context-answer pair.

**Distribution Matching-based Ranking.** In this ranking method, we fine-tune a separate language model to rank the overgenerated question candidates by predicting scores over these generated questions with a similar distribution to the ROUGE-L scores between the generated questions and the ground truth question. This distribution matching objective encourages the ranking language model to associate higher scores with questions similar to the ground truth question written by human education experts. We select the question with the highest score predicted by the ranking model as the best question for the given context-answer pair. Our method inspired from (Shi et al., 2023) trains a ranking language model to minimize the KL divergence (Joyce, 2011) between the distribution of the model-predicted scores over the generated questions and the distribution of ROUGE-L scores computing similarity of the generated questions to the human educator-written ground truth question. We outline the training process of the ranking model in Figure 2.

More specifically, we use a pre-trained ConvBERT (Jiang et al., 2020) model as our ranking language model. We use a combination of the given context-answer pair and the generated question to rank as input to the model. We feed the [CLS] embedding vector to a learnable linear layer during fine-tuning. For the  $i^{\text{th}}$  training question,  $P_\phi(\hat{q}_i) \in [0, 1]^K$  denotes the probability distribution of the model-predicted scores for generated questions and  $R(\hat{q}_i, q_i) \in [0, 1]^K$  denotes the probability distribution of the ROUGE-L scores

between the generated questions and the ground-truth question. Equation 2 shows the fine-tuning objective of the ranking language model to minimize the KL divergence between the model-predicted score distribution and the ROUGE-L score distribution. The softmax in equation 3 computes the distribution of the model-predicted scores where  $\phi(\hat{q}_{i,j}, c_i, a_i)$  denotes the score predicted by the ranking language model for the  $j^{\text{th}}$  generated question  $\hat{q}_{i,j}$  corresponding to the  $i^{\text{th}}$  context-answer pair  $(c_i, a_i)$ . The softmax in equation 4 computes the probability distribution of the ROUGE-L scores where  $r(\hat{q}_{i,j}, q_i)$  denotes the ROUGE-L score between the  $j^{\text{th}}$  generated question  $\hat{q}_{i,j}$  and the ground-truth question  $q_i$ . The hyperparameters  $\alpha_P$  and  $\alpha_R$  control the temperature of the softmax over the model-predicted scores and the ROUGE-L scores, respectively. The optimization problem is formally written as:

$$\text{minimize}_{\phi} \quad \frac{1}{N} \sum_i^KL(P_{\phi}(\hat{q}_i)||R(\hat{q}_i, q_i)), \quad (2)$$

$$\text{where } [P_{\phi}(\hat{q}_i)]_j = \frac{\exp \alpha_P \cdot \phi(\hat{q}_{i,j}, c_i, a_i)}{\sum_j \exp \alpha_P \cdot \phi(\hat{q}_{i,j}, c_i, a_i)}, \quad (3)$$

$$[R(\hat{q}_i, q_i)]_j = \frac{\exp \alpha_R \cdot r(\hat{q}_{i,j}, q_i)}{\sum_j \exp \alpha_R \cdot r(\hat{q}_{i,j}, q_i)}. \quad (4)$$

## 4 Experimental Evaluation

In this section, we describe the experimental setup to validate the effectiveness of our question generation methods.

### 4.1 Metrics and Baselines

To compare with prior work (Xu et al., 2022b), we use the ROUGE-L F1 score (Lin, 2004) (referred to as ROUGE-L) to evaluate the quality of generated questions. We compare our question generation methods to the existing state-of-the-art baseline (Xu et al., 2022b) which fine-tunes a BART LM (Lewis et al., 2020) to generate the ground truth question conditioned on the given context-answer pair.

### 4.2 Implementation Details

We use a pre-trained Flan-T5-Large model (Chung et al., 2022) with 770M parameters as our base LM for question generation; all implementation was done using the HuggingFace (Wolf et al., 2020) transformers library. We fine-tune the base LM for 10 epochs with early stopping on the validation loss

using the AdamW (Loshchilov and Hutter, 2017) optimizer with a learning rate of  $3e-4$  and a batch size of 8. Each epoch takes 20 minutes on a single NVIDIA A100 GPU.

FairytaleQA is imbalanced with respect to question attributes, with action and causal relationship accounting for 60% of the dataset. Our data augmentation method generates around 2500 synthetic questions over only the minority attributes: character, setting, feeling, outcome resolution, and prediction, to balance the training set. We fine-tune our base LM with the same setup described before on the augmented training set using a weight  $\lambda$  for the loss objective (see Equation 1) with original human educator-written questions and a different weight  $1 - \lambda$  for synthetic questions. Through a grid search, we find that setting  $\lambda = 0.8$  results in the best performance.

Our overgenerate-and-rank method generates question candidates using contrastive search (Su et al., 2022) (top-k of 4,  $\alpha$  penalty of 0.6) and nucleus sampling (Holtzman et al., 2020) (top-p of 0.9, temperature of 1) for perplexity-based ranking and distribution matching-based ranking, respectively. Through a grid search, we find that setting the softmax temperature hyperparameters as  $\alpha_P = 1e - 3$  and  $\alpha_R = 1e - 2$  results in the best performance.

## 5 Results and Discussion

**Overall Performance.** We report the average ROUGE-L across all test questions in the FairytaleQA dataset for all question generation methods in Table 2. The choice of the base language model is key when fine-tuning language models for question generation; fine-tuning Flan-T5 provides a significant improvement of 3.7% over the current state-of-the-art baseline of fine-tuning BART (Xu et al., 2022b), possibly because Flan-T5 is instruction fine-tuned on a large number of tasks relevant to both question answering and question generation. Our data augmentation method, which enriches the training set with diverse questions, further improves performance by 0.25% over fine-tuning Flan-T5 on the original training set. Among our overgenerate-and-rank methods, perplexity-based ranking and distribution matching-based ranking provide a 0.5% and 1.4% improvement over fine-tuning Flan-T5, respectively. Overall, our best method, distribution matching-based ranking method, provides a 5% absolute improvement over

Method	Questions		
	All	Explicit	Implicit
BART (Xu et al., 2022b)	0.5270	-	-
Flan-T5	0.5639	0.5998	0.4571
Data Augmentation	0.5664	0.5994	0.4682
Perplexity-based Ranking	0.5689	0.6057	0.4591
Distribution Matching-based Ranking	<b>0.5778</b>	<b>0.6107</b>	<b>0.4798</b>

Table 2: Experimental results on the FairytaleQA dataset in ROUGE-L (higher is better). Our methods significantly outperform existing baselines.

the current state-of-the-art BART baseline. This significant improvement shows that our data augmentation and overgenerate-and-rank methods are effective at making question-generation systems more robust, which results in better questions being generated. We also experiment with combining our data augmentation and overgenerate-and-rank methods. However, perhaps surprisingly, this combination does not lead to significant improvement in performance. We think that this result is possibly due to synthetic questions being too diverse in many cases with respect to the ground truth question. Therefore, controlling the diversity of synthetic questions for better alignment with those written by human educators is an important direction for future work.

### Performance Stratified by Question Category.

To gain more insight into the performance of our question generation methods, we also report the average ROUGE-L over test questions in the explicit and implicit categories. For the harder implicit questions with answers not explicitly included in the context as text spans, our data augmentation and distribution matching-based ranking methods improve performance by 1.2% and 2.3% over fine-tuning Flan-T5, respectively. This significant performance improvement shows that our data augmentation and overgenerate-and-rank methods are well-suited for harder question generation tasks, especially when given an answer that needs to be inferred from the context, for which the ground truth questions are already highly diverse.

**Data Augmentation Variants.** We report ROUGE-L scores for several variants of our data augmentation method in Table 4 in the Supplementary Material. FairytaleQA is imbalanced with respect to question attributes, with action and causal relationship accounting for 60% of the dataset. Augmenting all questions across all attributes results in a drop in performance. This observation validates our best data augmentation method, which is to generate synthetic questions for only the minority attributes: character, setting, feeling, outcome resolution, and prediction, to balance the training set. Moreover, fine-tuning Flan-T5 by weighting the human educator-written questions and synthetically-generated questions differently further improves performance.

**Different Decoding Strategies.** We report ROUGE-L scores for our overgenerate-and-rank methods combined with different choices of decoding strategy for overgeneration: greedy, nucleus sampling, and contrastive search, in Table 5 in the Supplementary Material. We compare perplexity-based ranking and two variants of distribution matching-based ranking trained on questions generated by nucleus sampling and contrastive search, respectively. We see that there is no single best decoding strategy that works across all ranking methods. We also observe that using the same decoding strategy for overgenerating candidate questions for both training and testing of the ranking method might not provide the best performance. For example, the distribution matching-based ranking method trained on questions generated by contrastive search works best at test time by ranking questions generated by nucleus sampling.

## 5.1 Qualitative Analysis

**Analysis of Questions Generated.** We provide a qualitative analysis of our question generation methods on an example context-answer-question triplet from the test set of FairytaleQA in Table 3. We observe that there are multiple relevant questions with different linguistic style and structure for the example context-answer pair; among them, our question generation methods need to generate the human educator-written ground truth question, “What did the man tell dullhead to do?”. Our fine-tuned Flan-T5 model generates a plausible but vague question excluding the subject of the context, “the old man”, that is not very similar to the ground truth question, possibly due to limitations

Context	... and when they had finished the little grey old man said to the dullhead: "Now I will bring you luck, because you have a kind heart and are willing to share what you have with others. there stands an old tree; cut it down, and amongst its roots, you'll find something." ...
Answer	cut down an old tree.
Ground truth question	What did the man tell dullhead to do?
Flan-T5	What did dullhead need to do to find something?
Data Augmentation	What did the little grey old man tell dullhead to do?
Perplexity-based Ranking	<ol style="list-style-type: none"> <li>1. What did the little man tell dullhead to do because he was willing to share what he had?</li> <li>2. What did the little man tell dullhead to do because he wanted to find something?</li> <li>3. What will dullhead do after he has eaten and drank the cake and beer?</li> <li>4. What will dullhead do to find something?</li> <li>5. What will dullhead do when he meets the grey old man?</li> </ol>
Distribution Matching-based Ranking	<ol style="list-style-type: none"> <li>1. What did the grey old man ask dullhead to do?</li> <li>2. What did the little grey old man say he wanted dullhead to do?</li> <li>3. What did the little man tell dullhead to do because he was willing to share what he had?</li> <li>4. What did the little man tell dullhead to do because he wanted to find something?</li> <li>5. What will dullhead need to do?</li> </ol>

Table 3: Qualitative analysis with an example input context-answer-question from the FairytaleQA dataset and question generated by our methods. Both data augmentation and overgenerate-and-rank improve diversity among the generated questions, which makes question generation more robust.

of greedy decoding. Our data augmentation method generates a much better question that is similar in structure and style to the ground truth question, which suggests that training on diverse questions is effective.

We also show the top five questions among the candidates, ranked by our overgenerate-and-rank methods. Our perplexity-based ranking method improves upon the fine-tuned Flan-T5 model by matching the structure of the ground truth question, "What did the man tell dullhead ...", but favors longer questions with more context information than the human educator-written question. Our distribution matching-based ranking method performs best by matching both the structure and style of the ground truth question. This example demonstrates that ranking methods trained on actual human preference information can be effective at identifying human-like questions among diverse candidates.

**Error Analysis.** We randomly select 30 context-answer pairs from the FairytaleQA test set with low ROGUE-L scores (less than 0.2) and investigate the questions generated by our best method, distribution matching-based ranking, and analyze why it does not perform well in these cases. We identify three main error types and list them in Table 7 in the Supplementary Material, with corresponding examples containing the input context-answer pair, the ground truth question, and the best generated question. The three main error types are: 1) character coreference resolution, 2) out-of-context

ground-truth questions, and 3) multiple evidence angles in the context.

The first two error types are beyond our control but the third type suggests that our methods have plenty of room for improvement. Errors of type character coreference resolution can occur when an input context has multiple characters and coreferences. In the first example, "self" is used as a complex coreference and confuses the question generation method. Errors of type out-of-context ground-truth questions can occur for ground-truth questions using information present outside the context the model sees as input. These ground-truth questions are human errors often referring to named entities present in other sections of the same story but not included in the input context. In the second example, the ground truth question refers to the character "Ian" who is not present in the context; the generated question uses the reference of "fisher's son" that is has access to in the given context. Errors of type multiple evidence angles can occur when the input context discusses different aspects of an answer. In the third example, the event of "Norseman invasion" in the answer could have questions related to either its cause, "people being wicked", or its timeline, "happening after the two Countesses fled to Scotland". As a result, among the top decoder output questions, there are none that discusses the latter, which is contained in the ground-truth question. Therefore, it is important to develop methods that can take all possible question

angles into account during decoding.

## 6 Conclusions and Future Work

In this paper, we proposed methods for improving automated answer-aware reading comprehension question generation by generating diverse question candidates and ranking them to align with human educator preferences. First, we proposed a data augmentation method that augments the training dataset with diverse questions obtained from a larger language model. Second, we proposed an overgenerate-and-rank method with two choices of ranking criterion, perplexity-based ranking and distribution matching-based ranking. The latter learns to rank the generated candidate questions to select ones that are closer to human-written questions. We conducted extensive experiments on the FairytaleQA dataset to validate the effectiveness of our methods showing that our best method provides an absolute improvement of 5% in ROUGE-L over the current state-of-the-art on this dataset. We also showed that our methods are significantly better than baselines in generating harder questions whose answers are not directly present in the context as text spans and have to be inferred.

There are several directions for future work. First, we can experiment with other data augmentation methods, e.g., by fine-tuning the base language model by weighting synthetically-generated questions according to their ROUGE-L scores with respect to the ground truth question. Second, we can explore the use of chain-of-thought (Wei et al., 2022) or self-ask (Press et al., 2022) to prompt the large language model in our data augmentation method. Third, we can experiment with other ranking objectives, such as ones using the Bradley-Terry model (Bradley and Terry, 1952) or ones using reinforcement learning with human feedback framework (Ziegler et al., 2019), to select the best questions that are aligned with human preference. Fourth, we can apply our methods to other question generation scenarios that require reasoning, such as logistical questions in online course discussion forums (Zylich et al., 2020), to help instructors anticipate common student questions.

## Acknowledgements

We thank the anonymous reviewers for their helpful comments. We thank the Learning Agency Lab for organizing the Quest for Quality Ques-

tions challenge<sup>3</sup> which inspired our initial work. We also thank Alexander Scarlatos and Naiming Liu for helpful discussions around this work. The authors also thank the NSF (under grants 1917713, 2118706, 2202506, 2215193, 2237676) for partially supporting this work.

## References

- Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.
- Liam Dugan, Eleni Miltsakaki, Shriyash Upadhyay, Etan Ginsberg, Hannah Gonzalez, DaHyeon Choi, Chuning Yuan, and Chris Callison-Burch. 2022. A feasibility study of answer-unaware question generation for education. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1919–1926.
- Fraide A Ganotice Jr, Kevin Downing, Teresa Mak, Barbara Chan, and Wai Yip Lee. 2017. Enhancing parent-child relationship through dialogic reading. *Educational Studies*, 43(1):51–66.
- Roberta Michnick Golinkoff, Erika Hoff, Meredith L Rowe, Catherine S Tamis-LeMonda, and Kathy Hirsh-Pasek. 2019. Language matters: Denying the existence of the 30-million-word gap has serious consequences. *Child development*, 90(3):985–992.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. [The curious case of neural text de-generation](#). In *International Conference on Learning Representations*.
- Zi-Hang Jiang, Weihao Yu, Daquan Zhou, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. 2020. Convbort: Improving bert with span-based dynamic convolution. *Advances in Neural Information Processing Systems*, 33:12837–12848.

<sup>3</sup><https://www.thequestchallenge.org>

- James M Joyce. 2011. Kullback-leibler divergence. In *International encyclopedia of statistical science*, pages 720–722. Springer.
- Tomáš Kočický, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.
- 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*, 30:121–204.
- Yash Kumar Lal, Nathanael Chambers, Raymond Mooney, and Niranjan Balasubramanian. 2021. [TellMeWhy: A dataset for answering why-questions in narratives](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 596–610, 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.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.
- Julie S Lynch, Paul Van Den Broek, Kathleen E Kremer, Panayiota Kendeou, Mary Jane White, and Elizabeth P Lorch. 2008. The development of narrative comprehension and its relation to other early reading skills. *Reading Psychology*, 29(4):327–365.
- Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*.
- 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. *The Journal of Machine Learning Research*, 21(1):5485–5551.
- Manav Rathod, Tony Tu, and Katherine Stasaski. 2022. Educational multi-question generation for reading comprehension. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 216–223.
- Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.
- Susan Sim and Donna Berthelsen. 2014. Shared book reading by parents with young children: Evidence-based practice. *Australasian Journal of Early Childhood*, 39(1):50–55.
- Katherine Stasaski, Manav Rathod, Tony Tu, Yunfang Xiao, and Marti A Hearst. 2021. Automatically generating cause-and-effect questions from passages. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 158–170.
- Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. [A contrastive framework for neural text generation](#). In *Advances in Neural Information Processing Systems*.
- 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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Zichao Wang, Andrew S Lan, Weili Nie, Andrew E Waters, Phillip J Grimaldi, and Richard G Baraniuk. 2018. 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.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.
- 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 *EMNLP: System Demonstrations*, pages 38–45.
- Guangxuan Xu, Paulina Toro Isaza, Moshi Li, Akin-toye Oloko, Bingsheng Yao, Aminat Adebeyi, Yunfang Hou, Nanyun Peng, and Dakuo Wang. 2022a. Nece: Narrative event chain extraction toolkit. *arXiv preprint arXiv:2208.08063*.



- Ying Xu, Dakuo Wang, Penelope Collins, Hyelim Lee, and Mark Warschauer. 2021. Same benefits, different communication patterns: Comparing children’s reading with a conversational agent vs. a human partner. *Computers & Education*, 161:104059.
- Ying Xu, Dakuo Wang, Mo Yu, Daniel Ritchie, Bingsheng Yao, Tongshuang Wu, Zheng Zhang, Toby Li, Nora Bradford, Branda Sun, Tran Hoang, Yisi Sang, Yufang Hou, Xiaojuan Ma, Diyi Yang, Nanyun Peng, Zhou Yu, and Mark Warschauer. 2022b. [Fantastic questions and where to find them: FairytaleQA – an authentic dataset for narrative comprehension](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 447–460, Dublin, Ireland. Association for Computational Linguistics.
- Bingsheng Yao, Dakuo Wang, Tongshuang Wu, Zheng Zhang, Toby Li, Mo Yu, and Ying Xu. 2022. [It is AI’s turn to ask humans a question: Question-answer pair generation for children’s story books](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 731–744, Dublin, Ireland. Association for Computational Linguistics.
- Xingdi Yuan, Tong Wang, Yen-Hsiang Wang, Emery Fine, Rania Abdelghani, Pauline Lucas, H el ene Sauz eon, and Pierre-Yves Oudeyer. 2022. Selecting better samples from pre-trained llms: A case study on question generation. *arXiv preprint arXiv:2209.11000*.
- Andrea A Zevenbergen and Grover J Whitehurst. 2003. Dialogic reading: A shared picture book reading intervention for preschoolers. *On reading books to children: Parents and teachers*, 177:200.
- Zheng Zhang, Ying Xu, Yanhao Wang, Bingsheng Yao, Daniel Ritchie, Tongshuang Wu, Mo Yu, Dakuo Wang, and Toby Jia-Jun Li. 2022. Storybuddy: A human-ai collaborative chatbot for parent-child interactive storytelling with flexible parental involvement. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–21.
- Zhenjie Zhao, Yufang Hou, Dakuo Wang, Mo Yu, Chengzhong Liu, and Xiaojuan Ma. 2022. [Educational question generation of children storybooks via question type distribution learning and event-centric summarization](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5073–5085, Dublin, Ireland. Association for Computational Linguistics.
- Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.
- Bowei Zou, Pengfei Li, Liangming Pan, and Aiti Aw. 2022. Automatic true/false question generation for educational purpose. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 61–70.
- Brian Zylich, Adam Viola, Brokk Toggerson, Lara Al-Hariri, and Andrew Lan. 2020. Exploring automated question answering methods for teaching assistance. In *Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part I 21*, pages 610–622. Springer.

## Supplementary Material

```

T ← {(c1, a1, q1), ..., (cN, aN, qN)};
for i ← 1 to N do
  {q̂i,1, ..., q̂i,M} ←
  GenQuesCodex((ci, ai, qi));
  āi ← GenAnsCodex((ci, qi));
  for j ← 1 to M do
    âi,j ← GenAnsCodex((ci, q̂i,j));
    if ROUGE(âi,j, ai) > 0.5 or
      ROUGE(âi,j, āi) > 0.5 then
      T ← T ∪ {(ci, ai, q̂i,j)};
    end
  end
end

```

**Algorithm 1:** Our automated data augmentation method which first generates question candidates and then filters them using consistency matching.

Decoding Type	Perplexity-based Ranking	Distribution Matching-based Ranking with Nucleus Sampling (0.95, 1, 10)	Distribution Matching-based Ranking with Contrastive Search (4, 0.6, 10)
Greedy (No ranking)	0.5639	0.5639	0.5639
Nucleus Sampling (0.9, 1, 10)	0.5664	<b>0.5778</b>	0.5657
Nucleus Sampling (0.95, 1, 10)	0.5618	0.5717	<b>0.5678</b>
Nucleus Sampling (0.95, 1, 75)	0.5671	0.5766	0.5638
Contrastive Search (4, 0.6, 10)	<b>0.5689</b>	0.5719	0.5647

Table 5: Experimental results on the FairytaleQA dataset in ROUGE-L (higher is better) comparing different decoding strategies across our overgenerate-and-rank methods. We denote Nucleus Sampling N questions with a p value of P and temperature of T as Nucleus Sampling (P, T, N) and Contrastive Search of N questions with a top-k of K and alpha penalty of A as Contrastive Search (K, A, N).

Data Augmentation Method Variant	ROUGE-L
No Augmentation	0.5639
All Questions	0.5499
Minority Questions	0.5607
Minority Questions + λ Weighting	<b>0.5664</b>

Table 4: Experimental results on the FairytaleQA dataset in ROUGE-L (higher is better) comparing different variants of our data augmentation method.

Context	Ground Answer	Truth	Ground Truth Question	Generated Question	Generated Answer of Generated Question	Generated Answer of Ground Truth Question
... and with that the rat laid a linen thread in the youth's hand. "Heaven be praised!", said the youth when he was up above once more. "I'll not go down there again in a hurry." But he held the thread in his hand and danced and sang as usual ...	excited		How did the youth feel when the rat allowed him to go above?	How did the youth feel when he had the linen thread in his hand?	happy	happy

Table 6: Our data augmentation method on an example context-answer pair from FairytaleQA. We use two reference answers for consistency matching. In this example, although the generated answer of generated question (happy) does not match the reference ground truth answer (excited), the generated question is still consistent and included in the augmented training set since the generated answer matches the alternate reference of generated answer of the ground truth question (happy).

Context	Answer	Ground Truth Question	Generated Question	Error Type
"What is your name?" asked the girl from underground. "Self is my name," said the woman. That seemed a curious name to the girl, and she once more began to pull the fire apart. Then the woman grew angry and began to scold, and built it all up again. Thus they went on for a good while; but at last, while they were in the midst of their pulling apart and building up of the fire, the woman upset the tar-barrel on the girl from underground. Then the latter screamed and ran away, crying: "Father, father! Self burned me!" "Nonsense, if self did it, then self must suffer for it!" came the answer from below the hill.	The girl.	Who did the girl's father think burned the girl?	Who screamed and ran away?	Character coreference resolution
So the gallows was built upon a high platform, and the fisher's son mounted the steps up to it, and turned at the top to make the speech that was expected from every doomed man, innocent or guilt. As he spoke he happened to raise his arm, and the king's daughter, who was there at her father's side, saw the name which she had written under it. With a shriek she sprang from her seat, and the eyes of the spectators were turned towards her. 'Stop! stop!' she cried, hardly knowing what she said. 'If that man is hanged there is not a soul in the kingdom but shall die also.' And running up to where the fisher's son was standing, she took him by the hand, saying, 'Father, this is no robber or murderer, but the victor in the three races, and he loosed the spells that were laid upon me.'	The king's daughter saw the name which she had written under it.	How did the princess recognize Ian?	What happened when the fisher's son raised his arm?	Out-of-context ground-truth questions
His vengeance was balked, however, for in the panic and confusion that followed Harold's death, the two Countesses slipped out of the Palace and fled to the coast, and took boat in haste to Scotland, where they had great possessions, and where they were much looked up to, and where no one would believe a word against them. But retribution fell on them in the end, as it always does fall, sooner or later, on everyone who is wicked, or selfish, or cruel; for the Norsemen invaded the land, and their Castle was set on fire, and they perished miserably in the flames. When Earl Paul found that they had escaped, he set out in hot haste for the Island of Hoy, for he was determined that the Dwarf, at least, should not escape. But when he came to the Dwarfie Stone he found it silent and deserted, all trace of its uncanny occupants having disappeared.	Norsemen invaded the land, and their Castle was set on fire, and they perished miserably in the flames.	What happened after the two Countesses fled to Scotland?	What happened because everyone who is wicked, or selfish, or cruel?	Multiple evidence angles in context

Table 7: Qualitative error analysis of our best method, distribution matching-based ranking, showing error examples from the FairytaleQA for each error type.

# Assisting Language Learners: Automated Trans-Lingual Definition Generation via Contrastive Prompt Learning

Hengyuan Zhang<sup>1</sup>, Dawei Li<sup>2</sup>, Yanran Li<sup>3†</sup>, Chenming Shang<sup>1</sup>, Chufan Shi<sup>1</sup>, Yong Jiang<sup>1</sup>

<sup>1</sup>Shenzhen International Graduate School, Tsinghua University

<sup>2</sup>Halicioğlu Data Science Institute, University of California, San Diego

<sup>3</sup>Independent Researcher

zhang-hy22@mails.tsinghua.edu.cn

## Abstract

The standard definition generation task requires to automatically produce mono-lingual definitions (e.g., English definitions for English words), but ignores that the generated definitions may also consist of unfamiliar words for language learners. In this work, we propose a novel task of **Trans-Lingual Definition Generation (TLDG)**, which aims to generate definitions in another language, i.e., the native speaker’s language. Initially, we explore the unsupervised manner of this task and build up a simple implementation of fine-tuning the multi-lingual machine translation model. Then, we develop two novel methods, Prompt Combination and Contrastive Prompt Learning, for further enhancing the quality of the generation. Our methods are evaluated against the baseline Pipeline method in both rich- and low-resource settings, and we empirically establish its superiority in generating higher-quality trans-lingual definitions. The ablation studies and further analysis are also conducted to provide more hints on this new task.

## 1 Introduction

A significant area of research within Intelligent Computer-Assisted Language Learning (ICALL) is devoted to supporting language learners in understanding words (Enayati and Gilakjani, 2020; Lolita et al., 2020). This research is primarily motivated by two main issues: (1) Language learners often struggle to accurately identify the meaning of words with multiple definitions, as the cognitive process of differentiating each meaning can be challenging (Tyler and Evans, 2001); (2) On another note, lexicographers are responsible for manually updating predefined word-definition inventories for language learners, a process that may be time-consuming and not always able to keep up with the constantly evolving nature of language usage. To address these issues, researchers aim to

<sup>†</sup> Corresponding author

### Chinese native speaker learning English words



Word: double

Context: ate a double portion.

Generated definition: 形容数值翻倍增加。

(Describing a numerical value that increases by a factor of two.)



### English native speaker learning Chinese words



Word: 近

Context: 包公庙... , 近因电视剧包青天脍炙人口而引来参拜人潮。

(The Bao Gong Temple..., has recently attracted crowds of worshippers due to the popular TV drama “Bao Qintian”.)

Generated definition: to indicate that not a long time ago.



Figure 1: The application scenes of a Chinese native speaker learning English and English native speaker learning Chinese. We also build a Chrome extension (in Appendix A) to better show the application scenes.

benefit both language learners and lexicographers by automatically generating the definition for a given word based on its corresponding local context (Ni and Wang, 2017; Gadetsky et al., 2018; Ishiwatari et al., 2019; Bevilacqua et al., 2020).

Previous works on definition generation mainly focus on mono-lingual generation scenarios, primarily due to the availability of parallel training and evaluation data (Yang et al., 2020; Huang et al., 2021; Zhang et al., 2022a). However, these works rarely notice a real-occurring problem that the generated definitions may also consist of unfamiliar words for language learners (Zhang, 2011). In other words, it is more applicable to generate definitions in the native language of foreign learners. As depicted in Figure 1, if a Chinese native speaker wants to know an English word’s meaning, the definition in Chinese is easier to capture.

To this end, we propose a novel task called **Trans-Lingual Definition Generation (TLDG)**. The TLDG task is challenging because there are no trans-lingual parallel datasets, e.g., the word and context are in Chinese, and the definition is

Context	Generated Definition	Error Type
This food <u>revitalized</u> the patient.	食物使病人恢复活力。(Food revitalizes patients)	Ignore-task error
..., 各家各派对人性的看法极为不同。(..., Each party has a very different view of human nature.)	形容 (Describe) a person’s opinion about something.	Language-mix error

Table 1: Zh-En and En-Zh examples of the two error types in the unsupervised TLDG task. The target words are marked with underline in context.

in English. Also, building trans-lingual parallel datasets is labor-consuming. To address this, we leverage the data resources of mono-lingual definition generation and utilize translation model to explore the trans-lingual definition generation task in an unsupervised manner. During preliminary experiments, we find two typical types of errors in the generated results. As shown in Table 1<sup>1</sup>, *Ignore-task error* means the model only translates the input’s context but neglects the definition generation task. *Language-mix error* means words in different languages simultaneously appear in the generated definition.

To mitigate the problems, we develop two novel learning methods. For the Ignore-task error, we get inspired from task-oriented prompt learning (Chung et al., 2022; Akyürek et al., 2022), and design Prompt Combination method to force the models focus on generating trans-lingual definition rather than mere translation. In addition, we propose Contrastive Prompt Learning method based on an contrastive loss (Hadsell et al., 2006; Schroff et al., 2015), which separates language information from the task prompt and in turn acquires a better task prompt representation for definition generation. Due to the scarcity of definition generation data in numerous languages, we carry out extensive experiments in both rich- and low-resource situations. We demonstrate that the Contrastive Prompt Learning method is effective in addressing the two errors and capable of yielding higher-quality definitions when compared to the baselines in both scenarios.

In general, our contributions are as follows:

- To better assist language learners, we propose the task of TLDG in an unsupervised manner and identify two typical errors.

<sup>1</sup>In this paper, Zh-En means the input’s word and context are in Chinese, and the expected generated definition is in English. Other language combinations are also similar.

- We develop several methods to mitigate the problems and demonstrate the Contrastive Prompt Learning method yields promising performances in both rich- and low-resource scenarios.
- We analyze the methods through ablated and case studies, and provide several hints on this newly introduced task. Also, we build a Chrome extension (in Appendix A) to further show the application scene of our proposed task.

## 2 Related Work

### 2.1 Definition Generation

The task of definition generation is first proposed by Noraset et al. (2017), which aims to generate definitions from corresponding word embeddings. Subsequent studies have investigated a broader range of application scenarios and model architectures for generating definitions. To generate appropriate definitions for polysemies, Ni and Wang (2017) first introduce the context and input the context with the target word to a bi-encoder model. Following them, Ishiwatari et al. (2019) develop a method that incorporates a gate mechanism in the decoding stage to integrate the information of the word and context. There are also some works that try to model the semantic representation in a more detailed way. Specifically, they break down the meaning of the target word into several components and provide a fine-grained word representation for the generation stage (Li et al., 2020; Reid et al., 2020a).

Recently, some works adopt pre-trained encoder-decoder models in definition generation and achieved great success. Huang et al. (2021) use a re-ranking strategy to obtain proper specific definitions. Zhang et al. (2022a) regard word and definition as a semantic equivalence pair to do contrastive learning. However, all the aforementioned works focus on improving the quality of the generated definitions, and the difficulty of understanding the definition itself for language learners has been ignored.

Although Kong et al. (2022a) design a multi-task framework to generate definitions with more simple words, we argue that other factors like language grammar will still hinder language learners to understand the definition. To mitigate it, we propose a novel task of trans-lingual definition generation

to generate definitions in the target language.

## 2.2 Prompt Learning

In recent years, numerous pre-trained models have been introduced, e.g., GPT (Radford et al., 2018), BART (Lewis et al., 2019). To adapt these models for different downstream tasks, prompt learning has been widely used. Schick and Schütze (2020a) manually design discrete template prompts to transform the downstream task into the text-infilling task, which is closer to the pre-trained paradigm. Besides, in the conditional text generation field, both Zhang et al. (2022b) and Xie et al. (2022) regard attribute keywords as hard prompts and fuse them into the model to control the generation result. However, Manually designing hard prompts can be both tedious and challenging, later works suggest using the soft prompts that consist of multiple learnable embeddings for the downstream tasks (Li and Liang, 2021; Liu et al., 2021; Han et al., 2022).

Furthermore, some works propose that rather than updating the entire PLM, it is more effective to fix its parameters and only update the soft prompts (Lester et al., 2021; Qin and Eisner, 2021a). When using large PLMs as the backbone, this method can achieve comparable results to fine-tuning the entire model. In the low-resource scenario, Gu et al. (2021) apply prompt initialization and use several tasks to obtain generalized prompts for different downstream tasks. Zheng and Huang (2021) and Zhang et al. (2021) use the prompt learning strategy to get different task-oriented prompts with corresponding task-specific objectives and achieve satisfactory results.

In this work, we use prompt learning to indicate the task and address the Ignore-task error. By developing a novel contrastive prompt learning loss, we finally achieve promising performances on both rich- and low-resource TLDG.

## 3 Method

One straightforward approach to generating trans-lingual definitions is to develop a pipeline that initially produces mono-lingual definitions and then translates them into the desired language. This intuitive approach serves as one naive baseline, which we elaborate in the experiment section (Section 4.2).

Besides, in this section, we introduce 3 methods to better fit our task: (1) a simple implementation of fine-tuning on multi-lingual translation model; (2)

Prompt Combination method; and (3) Contrastive Prompt Learning method.

### 3.1 Task Formulation

The standard definition generation (DG) task is to generate the definition  $D = \{d_0, \dots, d_t\}$  for a given word or phrase  $W = \{w_i, \dots, w_j\}$  and its corresponding context  $C = \{w_0, \dots, w_k\}$  ( $0 < i < j < k$ ). Here, the context is a sentence containing the word. Note that standard DG is a mono-lingual task where the word, context, and definition are in the same language.

Distinguishedly, the task of trans-lingual definition generation (TLDG) is to generate trans-lingual definition  $D_{l_j}$  in language  $l_j$  for a given word  $W_{l_i}$  and context  $C_{l_i}$  in another language  $l_i$ . Since there does not exist TLDG example triplets  $\{(W_{l_i}, C_{l_i}, D_{l_j})\}$ , the only available resources are mono-lingual definition generation datasets. Hence, the TLDG task in this work can be regarded as a fully unsupervised task.

### 3.2 Simple Implementation of Directly Fine-tuning Translation Model

The newly introduced TLDG task aims to generate the trans-lingual definition without supervised parallel datasets. As neural machine translation (NMT) shows powerful performance in translation, as a preliminary attempt, we directly fine-tune multi-lingual NMT with existing mono-lingual DG datasets ( $G$ ). Concretely, we concatenate language prompt (which is predefined in the multi-lingual NMT model to specify the source and target languages), target word, and context ( $[L_{l_i}; W_{l_i}; C_{l_i}]$ ) as input  $X_{l_i}$  to the encoder. Similarly, we concatenate language prompt and definition ( $[L_{l_i}; D_{l_i}]$ ) as ground-truth  $Y_{l_i}$  to train the model, which can be formulated as:

$$P(Y_{l_i}|X_{l_i}) = \prod_t p(y_t|y_{<t}, X_{l_i}; \theta) \quad (1)$$

where  $y_t$  is the  $t$ -th token of  $Y_{l_i}$ ,  $\theta$  is the model’s parameters to be tuned. To optimize, a cross-entropy loss is utilized to assess the difference between the distribution generated by the model and the ground-truth distribution, and the loss function is as follows:

$$\mathcal{L}_{MLE} = - \sum_{\substack{(W_{l_i}, C_{l_i}, D_{l_i}) \in G_{l_i}, \\ l_i \in L}} \log P(Y_{l_i}|X_{l_i}; \theta) \quad (2)$$

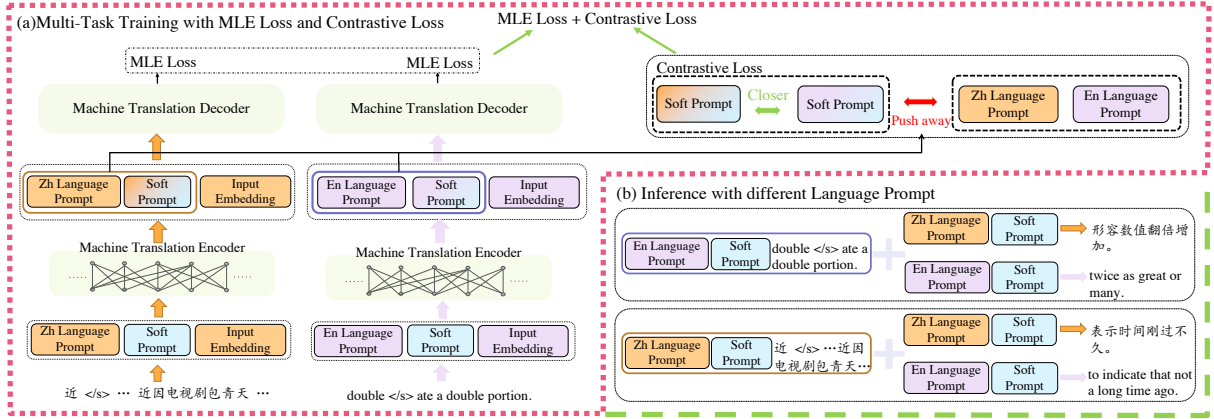


Figure 2: The area surrounded by the red dotted line represents the training process and the green dash line represents the inference process. In the training phase, (1) the task prompt will mix the language information from the language prompt (in blended color), and (2) the contrastive loss (upper right corner) is together applied with the MLE loss (upper left corner) to train the model jointly. At the inference stage, the language prompt could be set to any other languages used in the training stage for trans-lingual definition generation. Best viewed in color.

By concatenating the corresponding language prompt, the model is able to infer trans-lingual definitions in any language previously seen in the training stage ( $\langle W_{l_i}, C_{l_i} \rangle \rightarrow D_{l_j}, l_i, l_j \in L$ ).

### 3.3 Prompt Combination

Despite that fine-tuning the translation model seems plausible for trans-lingual definition generation, we find a plethora of Ignore-task cases in the generated definitions. We conjecture that the language prompt would still instinctively induce the translation model to perform the translation task, and thus leading to those Ignore-task errors.

To notify the model focus on the definition generation task, we add a specific task-oriented prompt after the language prompt. We adopt soft prompts for our task since they have been shown more flexible than hard prompts (Liu et al., 2022). In the training stage, we insert the task prompt  $T = \{t_1, t_2, \dots, t_n\}$  after the language prompt  $L_{l_i}$  for both encoder and decoder inputs, where  $n$  is the number of soft prompt tokens.

While this strategy mitigates the Ignore-task error in trans-lingual definition generation, we find adding the task-oriented soft prompt will lead to Language-mix errors. One possible explanation is that during the training stage, the task prompt is mixed up with the language information from the language prompt. During inference, such mixed task prompts will confuse the model to generate words in undesired languages.

### 3.4 Contrastive Prompt Learning

To tackle this problem, we propose a Contrastive Prompt Learning method. This method aims to obtain a more informative and representative task prompt by decoupling the language information inside within it. The overview of the proposed method is illustrated in Figure 2, where we take Chinese and English as examples.

In each batch, we randomly fetch training samples in two different languages ( $l_i$  and  $l_j$ ) and separate them into two groups. After passing each group into the model, we extract the language prompt embedding  $\mathbf{H}_{l_i}^{lp}$  and the task prompt embedding  $\mathbf{H}_{l_i}^{tp}$  from each group's encoding  $\mathbf{H}_{l_i}$  and  $\mathbf{H}_{l_j}$  according to their positions:

$$\mathbf{H}_{l_i}^{tp}, \mathbf{H}_{l_i}^{lp} = \text{Extract}(\mathbf{H}_{l_i}) \quad (3)$$

$$\mathbf{H}_{l_j}^{tp}, \mathbf{H}_{l_j}^{lp} = \text{Extract}(\mathbf{H}_{l_j}) \quad (4)$$

Since the language prompt only has one token, we directly regard language prompt embedding as language prompt representation  $\mathbf{h}_{l_i}^{lp}$ . For multiple task prompt tokens, we apply the pooling function to  $\mathbf{H}_{l_i}^{tp}$  and  $\mathbf{H}_{l_j}^{tp}$  to get the task prompt representation  $\mathbf{h}_{l_i}^{tp}$  and  $\mathbf{h}_{l_j}^{tp}$ . Without loss of generality, we implement attention-pooling, mean-pooling and max-pooling, and compare them in Section 4.5.

To build up contrastive loss, we regard task prompt representation in different languages as positive pairs  $(\mathbf{h}_{l_i}^{tp}, \mathbf{h}_{l_j}^{tp})$ , task prompt representation and different language prompt representation as negative pairs  $\{(\mathbf{h}_{l_i}^{tp}, \mathbf{h}_{l_j}^{lp}), l_i, l_j \in L\}$ . By doing

so, the language information in  $\mathbf{h}_{l_i}^{tp}$  and  $\mathbf{h}_{l_j}^{tp}$  can be effectively eliminated. Mathematically, the contrastive loss is formulated as:

$$\mathcal{L}_C = \max(d_p - d_n + \sigma, 0) / \tau \quad (5)$$

$$\begin{aligned} d_p &= \|\mathbf{h}_{l_i}^{tp} - \mathbf{h}_{l_j}^{tp}\| \\ d_n &= \sum_{a \in \{i, j\}} \frac{1}{2} \|\mathbf{h}_{l_i}^{tp} - \mathbf{h}_{l_a}^{lp}\| \end{aligned} \quad (6)$$

where  $d_p$  is the distance of positive pair,  $d_n$  is the average distance of negative pairs,  $\sigma$  is the margin and  $\tau$  is the temperature to scale the contrastive loss.

As Figure 2 depicts, the proposed contrastive loss is combined with MLE loss to train the model:

$$\mathcal{L}_{Final} = \lambda * \mathcal{L}_C + (1 - \lambda) * \mathcal{L}_{MLE} \quad (7)$$

where  $\lambda$  is a hyper-parameter to balance the two losses. In this way, our method is able to (1) separate the language information from the task prompt based on the novel contrastive loss, and (2) obtain a more oriented and pure task prompt representation for generating trans-lingual definition.

## 4 Experiments

In this section, we conduct extensive experiments and analyze the proposed methods carefully.

### 4.1 Datasets

Considering that many languages do not have sufficient definition generation data, we validate the proposed method in both rich- and low-resource scenarios. Note that all the datasets we use to train models are the mono-lingual definition generation datasets, which means the source language and target language are the same.

**Rich-resource** In the rich-resource scenario, we train and evaluate our models using English and Chinese definition generation datasets. For English, we use the Oxford dataset, collected using Oxford APIs of Oxford Dictionary<sup>2</sup> by Gadetsky et al. (2018). We follow Ishiwatari et al. (2019) to split them into training, validation, and test sets.

For Chinese, we follow Kong et al. (2022b) to use Chinese-WordNet (CWN) (Huang et al., 2010)

<sup>2</sup><https://developer.oxforddictionaries.com>

and split them into training, validation, and test sets. It is a semantic lexicon aiming to provide a knowledge base of sense<sup>3</sup>. The statistics of these two datasets are shown in Appendix B. In the inference stage, we conduct En-Zh, Zh-En trans-lingual definition generation.

**Low-resource** In the low-resource scenario, we set the training data size to 256, validation data size to 200, and following Schick and Schütze (2020b); Perez et al. (2021) to use the validation set as test set.

In specific, we build few-shot mono-lingual training datasets in English, Chinese, and France. For English and Chinese, we randomly choose samples from Oxford and CWN. For France, as there doesn't exist any public France definition generation dataset, we follow Reid et al. (2020b) to collect data from Lerobert Dictionary<sup>4</sup>. In the inference stage, we conduct trans-lingual definition generation with 6 settings, i.e., En-Zh, Zh-En, En-Fr, Fr-En, Zh-Fr, and Fr-Zh.

### 4.2 Experimental Settings

In this work, we utilize two multi-lingual NMT models, namely mBART-many-to-many<sup>5</sup> (a model that fine-tuned on mBART (Liu et al., 2020) with downstream machine translation tasks) and M2M<sup>6</sup> (a model that directly trained on massive multi-lingual translation tasks from scratch), to implement our ideas. For convenience, we use mBART-T to represent mBART-many-to-many in this paper.

For all experiments, we set the batch size to 16 and use Adam optimizer to update parameters. We train all of our models on a V100 GPU. Following Lester et al. (2021), we adopt 100 tunable soft prompt tokens. For the Contrastive Prompt Learning method, we set the temperature as 0.16 to scale the contrastive loss. The best performances in Section 4.4 adopt the attention-pooling function.

**Compared Methods** We compare with four methods: (1) A naive **Pipeline** method; (2) **Directly Fine-tuning** (Section 3.2); (3) **Prompt Combination** (Section 3.3); (4) **Contrastive Prompt** (Section 3.4). Specifically, the Pipeline method consists of generation and translation procedures. We begin with fine-tuning the pre-trained

<sup>3</sup><https://lope.linguistics.ntu.edu.tw/cwn2/>

<sup>4</sup><https://dictionnaire.lerobert.com>

<sup>5</sup><https://huggingface.co/facebook/mbart-large-50-many-to-many-mmt>

<sup>6</sup>[https://huggingface.co/facebook/m2m100\\_418M](https://huggingface.co/facebook/m2m100_418M)



Model	Method	Semantic Sim	
		En-Zh	Zh-En
mBART + M2M	Pipeline	47.58	52.24
M2M	w/ Directly Fine-tuning	45.69	51.68
	w/ Prompt Combination	47.56	52.63
	w/ Contrastive Prompt Learning	<b>49.42</b>	<b>55.19</b>
mBART-T	w/ Directly Fine-tuning	43.32	51.16
	w/ Prompt Combination	45.13	51.85
	w/ Contrastive Prompt Learning	47.79	53.92

Table 2: Automatic evaluation results on the rich-resource test dataset. The best results are in **bold**.

mBART (Liu et al., 2020) model with mono-lingual datasets to generate mono-lingual definitions rather than trans-lingual definitions. Subsequently, we utilize the M2M model to translate the generated definitions into the target language.

**Rich-resource** In the rich-resource scenario, we fine-tune all the parameters (including soft prompt tokens) of the model with 10 epochs. We set the learning rate  $5e-5$  for M2M, and  $1e-5$  for mBART-T and mBART.

**Low-resource** In the low-resource scenario, we use the prompt-tuning strategy only to tune the soft prompt tokens as suggested by Li and Liang (2021); Qin and Eisner (2021b). Following (Gu et al., 2021), we set training epochs to 30 and learning rate to  $1e-2$  for all models.

### 4.3 Evaluation Metrics

**Automatic Metrics** To measure the semantic quality of generated trans-lingual definitions, we apply the sentence-transformer toolkit (Reimers and Gurevych, 2020) to calculate the semantic similarity between the generated definition in the target language and the golden reference in its original language (e.g., for En-Zh, we calculate semantic similarity between generated Chinese definition and the golden English definition).

**Manual Evaluation** We also perform manual evaluation on the test set of 200 examples in low-resource setting. Based on the automatic evaluation results from Table 4.4, we only manually assess M2M model with three methods (Directly Fine-tuning, Prompt Combination, Prompt Contrastive Learning) in rich-resource setting, and M2M model with Prompt Contrastive Learning method in low-resource setting.

We ask six college students who achieved a score above 580 in the College English Test 6 level (CET-6) as annotators. Three of these students will be

responsible for annotating En-Zh results, while the remaining three will focus on Zh-En results. Similarly, we recruit six annotators who have passed Test national du français enseigné à titre de spécialité, niveau IV (TFS-4). Three of these annotators will be assigned to annotate En-Fr and Fr-En results, the remaining three will be responsible for Zh-Fr and Fr-Zh results.

Each annotator is asked to evaluate the generated trans-lingual definitions on two aspects: (1) Accuracy (Acc.) is a measure of the semantic relevance of the definitions to the word; (2) Fluency (Flu.) evaluates their readability without considering semantic alignment. Both criteria have a range of 1-5. In addition, the annotators are asked to rate the Ignore-task error and Language-mix error. We average the scores as the final score, and the agreements among the annotators of En-Zh, Zh-En, En-Fr & Fr-En, and Zh-Fr & Fr-Zh are ICC 0.937 ( $p < 0.001$ ), ICC 0.932 ( $p < 0.001$ ), ICC 0.904 ( $p < 0.001$ ) and 0.929 ( $p < 0.001$ ) respectively.

### 4.4 Main Results

We begin by examining the automatic evaluation results in rich-resource settings. As shown in Table 2, applying Contrastive Prompt Learning method on M2M and mBART-T models outperform other strategies across En-Zh and Zh-En scenarios. Furthermore, the baseline Pipeline method exhibits a performance degradation of 1.84 (En-Zh) and 2.95 (Zh-En) on the Semantic Sim metric when compared to our best method. This suggests that **the proposed Trans-lingual Definition Generation (TLDG) task cannot be simply addressed with a naive pipeline method**, which can be attributed to the errors accumulated during the pipeline.

Comparing the rows of M2M and mBART-T, M2M-based is superior on TLDG. We conjecture the superior performance comes from M2M’s translation ability, which is empirically validated in Fan et al. (2021). Since M2M model is trained

with massive parallel translation data and equipped with the Language-Specific Sparse technique, it is shown more powerful than mBART-T on translation tasks. The comparison between M2M and mBART-T gives us a hint that **model’s translation ability has an impact on our TLDG task**, which we analyze in later sections.

When checking the manual evaluation results in Table 3, it is notable that the proposed Contrastive Prompt Learning method obtains the highest scores on both Acc. and Flu. metrics. Comparing baseline Pipeline method with Contrastive Prompt Learning method in the Zh-En trans-lingual scenario (row 2 and row 8), we can see that Contrastive Prompt Learning method significantly improves trans-lingual quality, as it achieves 7.2% relative increase on Acc and 7.1% relative increase on Flu. A similar result in low-resource setting can refer to Appendix C.

Method	Language Combination	Acc. ↑	Flu. ↑
Pipeline (rich-resource)	En-Zh	3.09	3.34
	Zh-En	3.18	3.52
w/ Directly Fine-tuning (rich-resource)	En-Zh	3.02	3.37
	Zh-En	3.08	3.61
w/ Prompt Combination (rich-resource)	En-Zh	3.13	3.45
	Zh-En	3.17	3.67
w/ Contrastive Prompt (rich-resource)	En-Zh	<b>3.29</b> <sub>(+6.4%)</sub>	<b>3.51</b> <sub>(+5.1%)</sub>
	Zh-En	<b>3.41</b> <sub>(+7.2%)</sub>	<b>3.77</b> <sub>(+7.1%)</sub>
w/ Contrastive Prompt (low-resource)	En-Zh	2.98	3.31
	Zh-En	3.08	3.59
	En-Fr	3.04	3.48
	Zh-Fr	3.07	3.45
	Fr-En	3.11	3.62
	Fr-Zh	3.02	3.32

Table 3: Manual evaluation for quality assessment of trans-lingual definitions generated by M2M in low-resource test datasets

Another interesting finding comes when we compare the performances in rich- and low-resource scenarios. Take Zh-En trans-lingual task for example. It is observed that leveraging Contrastive Prompt Learning method in low-resource setting (row 10) is comparable to the simple implementation of directly fine-tuning (row 4) in rich-resource settings. Similar findings can also be found on the rows of En-Zh trans-lingual task. These findings greatly show the potential of the proposed method in the low-resource scenario. The results presented in Table 4 demonstrate that **our Contrastive Prompt Learning method effectively mitigates the two types of errors**. Specifically, when compared to directly fine-tuning implementa-

tion in the En-Zh scenario (row 1 and row 5), the Contrastive Prompt Learning method achieves a relative decrease of 77.8% in Language-mix error rate and perform well in Ignore-task error rate.

Method	Language Combination	Language-mix error rate↓	Ignore-task error rate↓
w/ Direct Fine-tuning (rich-resource)	En-Zh	-	11.25%
	Zh-En	-	9.50%
w/ Prompt Combination (rich-resource)	En-Zh	3.50%	7.50% <sub>(-33.3%)</sub>
	Zh-En	4.00%	6.00% <sub>(-36.8%)</sub>
w/ Contrastive Prompt (rich-resource)	En-Zh	-	2.50% <sub>(-77.8%)</sub>
	Zh-En	-	2.00% <sub>(-78.9%)</sub>

Table 4: Manual evaluation results of the two errors in trans-lingual definition generated by M2M in low-resource test datasets.

## 4.5 Ablation Study

**Pooling Function** To examine the variants of pooling functions as introduced in Section 3.4, we then conduct an ablation study on M2M model with the best task-ratio 0.2 obtained in Section 4.5.

As Table 5 shows, the attention-pooling function outperforms mean- and max- pooling functions on all the metrics. The reason lies in the distinctness of how these pooling functions gather token information. When constructing task prompt representation, the attention-pooling function aggregates all the task prompt tokens with the attention weight between the task and language prompt. Intuitively, the attention weight measures the degree of language information in each token of the task prompt. As a result, the task prompt representation based on attention-pooling contains more precise mixed language information, and in turn aids in separating language information when implementing Prompt Contrastive Learning. The variations observed in different pooling functions suggest that **the approach used to obtain an accurate representation is crucial in contrastive learning**.

Model & Method	Pooling Function	Semantic Sim	
		En-Zh	Zh-En
M2M		<b>49.42</b>	<b>55.19</b>
/w Contrastive Prompt	attention	48.91	54.75
/w Task Ratio 0.2	mean	48.83	54.68
	max		

Table 5: Ablation study results on the pooling functions. The best numbers are in **bold**.

**Hyper-Parameter** Another influential factor in our method is hyper-parameter  $\lambda$  in Eq. 7. To ex-

Model & Method	Task Ratio	Semantic Sim	
		En-Zh	Zh-En
M2M	0.1	48.81	55.12
/w Contrastive Prompt	0.2	<b>49.42</b>	<b>55.19</b>
/w Attention Pooling	0.3	48.24	54.76
	0.4	47.63	53.81
	0.5	47.87	53.79

Table 6: Hyper-parameter analysis results on the task ratio. The best results are in **bold**.

ploring its effect, we keep using attention-pooling in all settings and set different  $\lambda$  for each model to observe the performance change.

As Table 6 shows, when the task ratio is set to 0.2, the proposed method yields the best performance. When the task ratio is lower or higher than 0.2, the performances deteriorate. We conjecture that our model requires more generation loss to guide contrastive learning in the right way.

#### 4.6 Case Study

For better understanding, we present some cases under the rich-resource setting to vividly analyze the superiority of our Contrastive Prompt Learning method. Table 7 compares all methods on two trans-lingual scenarios. After examining the definitions produced by the directly fine-tuning implementation, we find undesired words like “经济” (*economy*) (in the En-Zh case), as well as the words “*interdependence*” and “*country*” (in the Zh-En case). All these words are the direct translations of the context words rather than the definitions. In the Zh-En case, it is clear that the definition from the Prompt Combination method contains Language-mix error, as it includes a Chinese word “形容” (*describe*). In the En-Zh case, the definition produced by the baseline Pipeline method includes an unsuitable explanation word “上升运动” (*upward movement*), which might be resulted from the limited definition style’s data in the translation model’s training corpus. In contrast, the Contrastive Prompt Learning method’s output, which includes “正面发展” (*positive development*) and “fewer or greater”, accurately represents the meaning of the target words. Drawing on the highest scores in Table 2 and Table 3, we safely conclude that the proposed **Prompt Contrastive Learning is more effective in trans-lingual definition generation**.

We also conduct case studies on the choice of multi-lingual translation model, as a complementary assessment to the results in Table 2. As shown in Table 8, the generated definitions of mBART-T

<i>Word</i>	upturn
<i>Context</i>	... in response to the economic upturn helped by a recovery of key western export markets.
<i>Pipeline</i>	某人或某物的状况中的上升运动 ( <i>The upward movement in the condition of someone or something.</i> )
<i>Directly Fine-tuning</i>	经济的好转。 ( <i>The improvement of the economy.</i> )
<i>Prompt Combination</i>	形容 ( <i>Describing</i> ) a rising trend of something.
<i>Contrastive Prompt</i>	比喻特定事件向正面发展。 ( <i>The specific event is developing towards a positive direction</i> )
<i>Word</i>	日益 ( <i>day by day</i> )
<i>Context</i>	... 各国相互依赖程度日益加深。 (... <i>the degree of interdependence among countries is increasingly deepening.</i> )
<i>Pipeline</i>	the degree is deepening.
<i>Directly Fine-tuning</i>	increasing interdependence of country.
<i>Prompt Combination</i>	in a gradual and increasing degree.
<i>Contrastive Prompt</i>	to an ever greater or fewer degree.

Table 7: Generated result comparison between four methods on M2M model.

<i>Word</i>	accent
<i>Context</i>	... cobalt blue was used to accent certain elements including ...
<i>M2M</i>	强调特定对象。 ( <i>Emphasize specific objects.</i> )
<i>mBART-T</i>	强调的重点。 ( <i>Key points to emphasize.</i> )
<i>Word</i>	珍惜 ( <i>cherish</i> )
<i>Context</i>	..., 什么又是值得你去珍惜的? (..., <i>what is worth cherishing for you?</i> )
<i>M2M</i>	deeply regard the value of something.
<i>mBART-T</i>	regard with great appreciation.

Table 8: Generated result comparison between M2M based and mBART-T based models.

contain “重点” (*key*) and “*appreciation*”, which are not accurate for explaining the corresponding words’ meanings. However, the M2M model handles these cases well. This case study further demonstrates the hint that **the translation capability of the backbone model is crucial for trans-lingual definition generation**. For more cases in both rich- and low-resource scenarios, please kindly refer to Appendix D.

## 5 Conclusions

In this work, we propose a novel and challenging task TLDG that generates the trans-lingual definition in an unsupervised manner. To tackle the task, we leverage multi-lingual translation models and propose an effective method of Contrastive Prompt Learning for the task. Through extensive experiments, we validate the method is capable of addressing typical errors and promising in both

rich- and low-resource scenarios. In the future, we will develop more strategies to improve the quality of trans-lingual definitions.

## Limitations

Our work has several limitations. In terms of method generalization, the proposed method depends on multi-lingual neural machine translation models to generate trans-lingual definitions, and hence limits its application scope to those languages rarely supported by translation models. Moreover, our findings are based on three languages, but different families of languages may exhibit distinct phenomenon that even challenges our conclusions.

## References

- Afra Feyza Akyürek, Sejin Paik, Muhammed Kocyigit, Seda Akbiyik, Serife Leman Runyun, and Derry Wijaya. 2022. [On measuring social biases in prompt-based multi-task learning](#). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 551–564, Seattle, United States. Association for Computational Linguistics.
- Michele Bevilacqua, Marco Maru, and Roberto Navigli. 2020. Generationary or “how we went beyond word sense inventories and learned to gloss”. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7207–7221.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.
- Fatemeh Enayati and Abbas Pourhosein Gilakjani. 2020. The impact of computer assisted language learning (call) on improving intermediate efl learners’ vocabulary learning. *International Journal of Language Education*, 4(1):96–112.
- Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22(107):1–48.
- Artyom Gadetsky, Ilya Yakubovskiy, and Dmitry Vetrov. 2018. Conditional generators of words definitions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 266–271.
- Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2021. Ppt: Pre-trained prompt tuning for few-shot learning. *arXiv preprint arXiv:2109.04332*.
- Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, volume 2, pages 1735–1742. IEEE.
- Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2022. Ptr: Prompt tuning with rules for text classification. *AI Open*, 3:182–192.
- Chu-Ren Huang, Shu-Kai Hsieh, Jia-Fei Hong, Yun-Zhu Chen, I-Li Su, Yong-Xiang Chen, and Sheng-Wei Huang. 2010. Chinese wordnet: Design, implementation, and application of an infrastructure for cross-lingual knowledge processing. *Journal of Chinese Information Processing*, 24(2):14–23.
- Han Huang, Tomoyuki Kajiwara, and Yuki Arase. 2021. Definition modelling for appropriate specificity. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2499–2509.
- Shonosuke Ishiwatari, Hiroaki Hayashi, Naoki Yoshinaga, Graham Neubig, Shoetsu Sato, Masashi Toyoda, and Masaru Kitsuregawa. 2019. Learning to describe unknown phrases with local and global contexts. 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 3467–3476.
- Cunliang Kong, Yun Chen, Hengyuan Zhang, Liner Yang, and Erhong Yang. 2022a. Multitasking framework for unsupervised simple definition generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5934–5943.
- Cunliang Kong, Yun Chen, Hengyuan Zhang, Liner Yang, and Erhong Yang. 2022b. [Multitasking framework for unsupervised simple definition generation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5934–5943, Dublin, Ireland. Association for Computational Linguistics.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.
- 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*.
- Jiahuan Li, Yu Bao, Shujian Huang, Xinyu Dai, and Jiajun Chen. 2020. Explicit semantic decomposition for definition generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 708–717.

- Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.
- Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68.
- Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.
- 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. *Transactions of the Association for Computational Linguistics*, 8:726–742.
- Yuri Lolita, Endry Boeriswati, and Ninuk Lustyantje. 2020. The impact of computer assisted language learning (call) use of english vocabulary enhancement. *Linguistic, English Education and Art (LEEA) Journal*, 4(1):206–221.
- Ke Ni and William Yang Wang. 2017. [Learning to explain non-standard English words and phrases](#). In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 413–417, Taipei, Taiwan. Asian Federation of Natural Language Processing.
- Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2017. Definition modeling: Learning to define word embeddings in natural language. In *Thirty-First AAAI Conference on Artificial Intelligence*.
- Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. *Advances in Neural Information Processing Systems*, 34:11054–11070.
- Guanghui Qin and Jason Eisner. 2021a. Learning how to ask: Querying lms with mixtures of soft prompts. *arXiv preprint arXiv:2104.06599*.
- Guanghui Qin and Jason Eisner. 2021b. Learning how to ask: Querying lms with mixtures of soft prompts. *arXiv preprint arXiv:2104.06599*.
- Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.
- Machel Reid, Edison Marrese-Taylor, and Yutaka Matsuo. 2020a. Vcdm: Leveraging variational bi-encoding and deep contextualized word representations for improved definition modeling. *arXiv preprint arXiv:2010.03124*.
- Machel Reid, Edison Marrese-Taylor, and Yutaka Matsuo. 2020b. Vcdm: Leveraging variational bi-encoding and deep contextualized word representations for improved definition modeling. *arXiv preprint arXiv:2010.03124*.
- Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. *arXiv preprint arXiv:2004.09813*.
- Timo Schick and Hinrich Schütze. 2020a. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*.
- Timo Schick and Hinrich Schütze. 2020b. It’s not just size that matters: Small language models are also few-shot learners. *arXiv preprint arXiv:2009.07118*.
- Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.
- Andrea Tyler and Vyvyan Evans. 2001. Reconsidering prepositional polysemy networks: The case of over. *Language*, pages 724–765.
- Yuqiang Xie, Yue Hu, Yunpeng Li, Guanqun Bi, Luxi Xing, and Wei Peng. 2022. Psychology-guided controllable story generation. *arXiv preprint arXiv:2210.07493*.
- Liner Yang, Cunliang Kong, Yun Chen, Yang Liu, Qinan Fan, and Erhong Yang. 2020. Incorporating sememes into chinese definition modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1669–1677.
- Hengyuan Zhang, Dawei Li, Shiping Yang, and Yanran Li. 2022a. Fine-grained contrastive learning for definition generation. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, pages 1001–1012.
- Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. 2021. Differentiable prompt makes pre-trained language models better few-shot learners. *arXiv preprint arXiv:2108.13161*.
- Yihua Zhang. 2011. Discussion on the definitions in chinese learner’s dictionaries: Comparative study of domestic and foreign learner dictionaries (translated from chinese). *Chinese Teaching in the World*.
- Zhexin Zhang, Jiaxin Wen, Jian Guan, and Minlie Huang. 2022b. Persona-guided planning for controlling the protagonist’s persona in story generation. *arXiv preprint arXiv:2204.10703*.

Chujie Zheng and Minlie Huang. 2021. Exploring prompt-based few-shot learning for grounded dialog generation. *arXiv preprint arXiv:2109.06513*.

## A Chrome Extension Application Scene

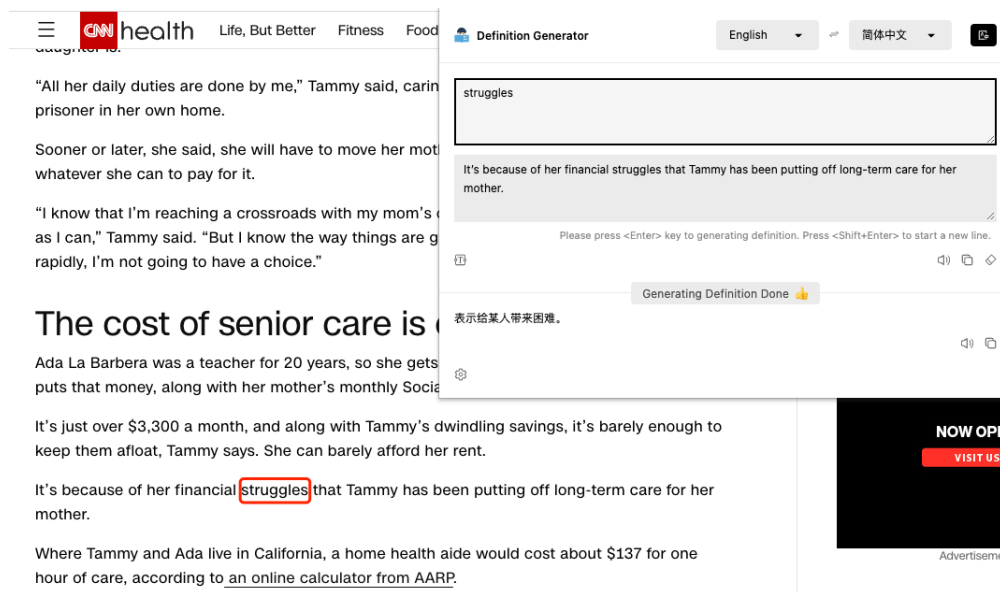


Figure 3: The application scene of Learning English words based on our best method. Given the word “struggles” and press the shortcut key, the application will identify its corresponding context and output the definition “表示给某人带来困难。” (To make someone difficult.).



Figure 4: The application scene of Learning Chinese words based on our best method. Select the word “辅” (supplement) and press the shortcut key, the application will identify its corresponding context and output the definition “Describing something as an accessory or auxiliary item”.

## B Rich-resource Detailed Dataset Setting

	Oxford			CWN		
	Train	Valid	Test	Train	Valid	Test
Words	33128	8,867	3881	6574	823	823
Entries	97,855	12,232	5111	67861	8082	8599
Context length	17.74	17.80	16.24	34.49	34.73	34.04
Desc. length	11.02	10.99	10.03	14.76	14.60	14.72

Table 9: Statistics of the Oxford (English) dataset and CWN (Chinese) dataset.

We use Oxford and CWN definition generation datasets in rich-resource setting experiment, the statistics of Oxford and CWN are shown in Table 9.

## C Human Evaluation of Pipeline method in Low-resource Setting

Language Combination	Method			
	/w Contrastive Prompt		Pipeline	
	Acc	Flu	Acc	Flu
En-Zh	2.98	3.31	2.73	3.09
Zh-En	3.08	3.59	2.91	3.34
En-Fr	3.04	3.48	2.88	3.31
Fr-En	3.11	3.62	2.98	3.46
Zh-Fr	3.07	3.45	2.92	3.27
Fr-Zh	3.02	3.32	2.74	3.13

Table 10: Human evaluation results of M2M /w Contrastive Prompt Learning method and baseline Pipeline method in low-resource setting.

We also compare our proposed M2M /w Contrastive Prompt Learning method with baseline Pipeline method in low-resource setting, the results are shown in Table 10.



## D Generated Results

### D.1 Rich-Resource Generated Results

<i>Word</i>	telex
<i>Context</i>	they telexed the company denying breach of contract.
<i>Generated Result</i>	以电传方式传送讯息。
<i>Word</i>	bulky
<i>Context</i>	radio could communicate between cities, but they were too bulky to be man-carried.
<i>Generated Result</i>	形容体积大的。
<i>Word</i>	concession
<i>Context</i>	a corona and one adverb of resignation - or is it concession?
<i>Generated Result</i>	承认或授权后述对象。
<i>Word</i>	electronic
<i>Context</i>	1987 was an early but fertile time for electronic dance music.
<i>Generated Result</i>	以电子方式进行演奏。
<i>Word</i>	spiral
<i>Context</i>	tensions have spiraled between pyongyang and the us.
<i>Generated Result</i>	比喻特定事件在一段长时间内持续进行。
<i>Word</i>	fortune
<i>Context</i>	I have had the good fortune to see the piece several times.
<i>Generated Result</i>	形容运气好。
<i>Word</i>	revitalize
<i>Context</i>	this food revitalized the patient.
<i>Prompt Combination</i>	使后述对象恢复生命力。
<i>Word</i>	意外
<i>Context</i>	好在我们都已买了保险，如果发生意外，一切都由保险公司理赔。
<i>Generated Result</i>	an unfortunate or unexpected occurrence of something.
<i>Word</i>	学术
<i>Context</i>	国立大学及所有私校没必要一窝蜂搞学术，现在学生所学和社会往往都是脱节的。
<i>Generated Result</i>	an academic activity of the university or community.
<i>Word</i>	立国
<i>Context</i>	立国精神、民族意识的观念如果不在军训课中提醒学生，根本没有机会。
<i>Generated Result</i>	the establishment of state.
<i>Word</i>	近
<i>Context</i>	包公庙..., 近因电视剧包青天脍炙人口而引来参拜人潮。
<i>Generated Result</i>	to indicate that not a long time ago.
<i>Word</i>	维
<i>Context</i>	怪手及人员到市场附近巡视，凡发现摊架，则一律予以铲除，以维公权力的威信。
<i>Generated Result</i>	maintain the state of ( something ).

Table 11: The generated results of M2M model with Contrastive Prompt Learning method under rich-resource setting.

## D.2 Low-Resource Generated Results

<i>Word</i>	concession
<i>Context</i>	a corona and one adverb of resignation - or is it concession?
<i>Generated Result</i>	形容被授权的。
<i>Word</i>	antithesis
<i>Context</i>	his theory is the antithesis of mine.
<i>Generated Result</i>	形容与特定事件相反的。
<i>Word</i>	conditional
<i>Context</i>	the conditional sale will not be complete until the full purchase price is paid.
<i>Generated Result</i>	形容有条件的。
<i>Word</i>	lame
<i>Context</i>	the comedy aspect is a little lame, with too many one-liners
<i>Generated Result</i>	形容缺乏活力的。
<i>Word</i>	surge
<i>Context</i>	the testing equipment-maker 's shares surged as sales rose for the first time in six quarters.
<i>Generated Result</i>	形容特定对象数量增加。
<i>Word</i>	近
<i>Context</i>	我认为太阳在清早刚出来的时候离人近，中午的时候离人远。
<i>Generated Result</i>	close to or nearby.
<i>Word</i>	意外
<i>Context</i>	好在我们都已买了保险，如果发生意外，一切都由保险公司理赔。
<i>Generated Result</i>	an accidental occurrence.
<i>Word</i>	看法
<i>Context</i>	我希望七月初开院士会议时，能够再提出在这方面一些具体的看法。
<i>Generated Result</i>	the opinion of a person.
<i>Word</i>	终究
<i>Context</i>	走在错误的路上，终究是要输的。
<i>Generated Result</i>	Décrivez le résultat final de l'événement.
<i>Word</i>	日益
<i>Context</i>	融入实际生活的经验中，人生经验便日益丰富。
<i>Generated Result</i>	Le degré de description est approfondi.
<i>Word</i>	revitalize
<i>Context</i>	this food revitalized the patient.
<i>Generated Result</i>	Donner une nouvelle vitalité.
<i>Word</i>	enter
<i>Context</i>	enter a drug treatment program.
<i>Generated Result</i>	Participer à un programme ou un projet.

Table 12: The generated results of M2M model with Contrastive Prompt Learning method under low-resource setting.

# Predicting the Quality of Revisions in Argumentative Writing

Zhexiong Liu<sup>1</sup>, Diane Litman<sup>1,2</sup>, Elaine Wang<sup>3</sup>, Lindsay Matsumura<sup>2</sup>, Richard Correnti<sup>2</sup>

<sup>1</sup>Department of Computer Science

<sup>2</sup>Learning Research and Development Center

University of Pittsburgh, Pittsburgh, Pennsylvania 15260 USA

<sup>3</sup>RAND Corporation, Pittsburgh, Pennsylvania 15213 USA

zhexiong@cs.pitt.edu, ewang@rand.org

{dlitman, lclare, rcorrent}@pitt.edu

## Abstract

The ability to revise in response to feedback is critical to students' writing success. In the case of argument writing in specific, identifying whether an argument revision (AR) is successful or not is a complex problem because AR quality is dependent on the overall content of an argument. For example, adding the same evidence sentence could strengthen or weaken existing claims in different argument contexts (ACs). To address this issue we developed Chain-of-Thought prompts to facilitate ChatGPT-generated ACs for AR quality predictions. The experiments on two corpora, our annotated *elementary essays* and existing *college essays* benchmark, demonstrate the superiority of the proposed ACs over baselines.

## 1 Introduction

Argumentative Revision (AR) in response to feedback is important for improving the quality of students' written work. Successful ARs<sup>1</sup> usually include adding relevant evidence, deleting repeated evidence or reasoning, and elaborating relevant evidence examples to support claims (Afrin et al., 2020). Differentiating between successful versus unsuccessful ARs, however, is a complex endeavor. For example, making the same AR in distinct Argumentative Contexts (ACs) could differentially affect the quality of a student's essay. Here the ACs are defined as pieces of sentences that present reasons, evidence, and claims supporting or opposing arguments in argumentative writing (see Sec. 4.2). For example, Figure 1 shows two pieces of ARs that added the same sentence “*it was hard for them to concentrate though, as there was no midday meal*” but caused opposite AR quality.

Recently developed Automated Writing Evaluation (AWE) systems have focused on assessing the content and structure of student essays to automatically provide students with formative feed-

<sup>1</sup>Afrin and Litman (2023) use the term desirable revisions.

### AR #372: Unsuccessful Revision

<original draft> **They also did not concentrate good because they did not have lunch over there.**

According to the text, many kids in Sauri did not attend school because their parents could not afford school fees. </original draft> <adding> **It was hard for them to concentrate though, as there was no midday meal.** </adding>

### AR #592: Successful Revision

<original draft> In 2010 the schools had minimal supplies like books, paper, and pencils, but the students wanted to learn. All of them worked hard with few supplies they had. </original draft>

<adding> **It was hard for them to concentrate though, as there was no midday meal.** </adding>

Figure 1: Two pieces of ARs in two student essays show that *adding* the same sentence “*it was hard for them to concentrate though, as there was no midday meal*” (bold in red) in different contexts caused opposite AR quality. AR #372 added a piece of evidence that *already existed* in the original draft (bold in blue) thus the attempted AR did not improve the essay quality. AR #592 improved the quality by adding a *relevant* piece of new evidence. AR #372 was *unsuccessful* while AR #592 was *successful*.

back (Zhang et al., 2016; Writing Mentor, 2016; Wang et al., 2020; Beigman Klebanov and Madnani, 2020). Successful revisions (e.g., adding relevant evidence) improve an essay's quality. Unsuccessful revisions, in contrast, lead to no improvement or can even weaken an essay's argument (Afrin et al., 2020). As a result, assessing the success of ARs is important to assess the quality of ARs in line with provided feedback.

AR quality has previously been predicted by using long and short neighboring contexts of ARs (Afrin and Litman, 2023). This location-based approach for constructing ACs did not exploit any argumentative relationships between ARs and potential ACs. Another study (Zhang and Litman,

2016) incorporated AR contexts with cohesion blocks and employed sequence labeling to model AR interdependence across revisions. This work predicted AR purposes from discourse structures but did not further study AR quality or analyze AR quality from the perspective of ACs. To bridge these gaps, we address three research questions. **RQ1:** To what extent are ACs helpful for predicting AR quality? **RQ2:** What type of AC is the most helpful in AR quality predictions? **RQ3:** Can ChatGPT prompts be used to generate useful ACs? In studying the three RQs, we have made the following contributions:

- Our project is the first in the revision field to analyze the relationship between ACs and AR quality predictions.
- We are among the first to incorporate the state-of-the-art large language model ChatGPT in generating ACs in argumentative writing.
- Experiments using both elementary and college essay corpora show the superiority of the proposed ACs over existing location-based contexts for AR quality predictions.

## 2 Related Work

### 2.1 Argumentative Revision in NLP

Revision research has been conducted using multiple types of Natural Language Processing (NLP) corpora ranging from Wikipedia to argumentative essays. While argumentative writing research has analyzed argumentative roles and discourse elements in persuasive writing (Stab and Gurevych, 2014; Song et al., 2020; Putra et al., 2021) (e.g., by studying the stance towards some topic, backing up claims, or following argumentative and rhetorical considerations), such analyses have not typically been applied to revision research in this domain. Revision research, in contrast, has primarily focused on grammar correction, paraphrasing, semantic editing (Yang et al., 2017), and analyzing revision purposes (Zhang and Litman, 2015; Shibani et al., 2018; Afrin et al., 2020; Kashefi et al., 2022). Although revision research has sometimes leveraged contextualized features during classification, the contextual features have been location-based (Zhang and Litman, 2016; Afrin and Litman, 2023). We instead extract contextual information from an essay based on argumentative essay analysis rather than on adjacency to a revision.

### 2.2 LLM in Argumentative Revision

Large Language Models (LLMs) have scaled up model sizes from a few million to hundreds of billions of parameters. Their strong capabilities of handling multiple downstream NLP tasks have made LLMs favorable in recent research (Chowdhery et al., 2022). Prior revision works, e.g., academic writing (Ito et al., 2019), debate assessment (Skitalinskaya et al., 2021), paraphrase generation (Mu and Lim, 2022), were mostly based on Transformer (Vaswani et al., 2017) and BERT (Devlin et al., 2019) models but not the cutting-edge LLMs, e.g., ChatGPT<sup>2</sup>. Pretrained LLMs have shown strong few-shot learning capabilities by way of developing prompts to guide LLMs in generating successful outputs (Brown et al., 2020; Liu et al., 2023). For example, Chain-of-Thought (CoT) prompts (Wei et al., 2022) enable pretrained LLMs to solve complex reasoning problems by decomposing the tasks into a series of intermediate steps. Kojima et al. (2022); Wang et al. (2022) investigated the effectiveness of CoT in multi-step reasoning, however, little work has used CoT for extracting and then generating tasks in the revision field. In this work, we leverage ChatGPT with CoT prompts to generate ACs in argumentative writing.

## 3 Corpora

### 3.1 Data Collection

AR corpora are rarely annotated in the revision community because of their expensive annotation costs. The publicly available *college essay* corpus for AR quality predictions (Afrin and Litman, 2023) contains paired drafts of argumentative essays written in response to an essay prompt (original drafts) and revised based on feedback (revised drafts). The corpus is comprised of 60 essays (N=60 college students), inclusive of both native and proficient non-native speakers of English, in response to an essay prompt about Technology Proliferation. Students received general feedback upon completion of their first drafts, asking them to add more examples in their second drafts. The second drafts then received non-textual feedback through the ArgRewrite system (Zhang et al., 2016) to help students write their third drafts. Afterward, the second and third drafts were collected as pairs of original and revised drafts.

We followed a similar protocol to collect 596 *ele-*

<sup>2</sup><https://openai.com/blog/chatgpt>

		Elementary Essays			College Essays		
		Reasoning	Evidence	Total	Reasoning	Evidence	Total
Successful	Add	769	671	1440	104	23	127
	Delete	213	104	317	7	1	8
	Modify	129	104	233	3	0	3
Unsuccessful	Add	360	491	851	87	2	89
	Delete	102	147	249	6	0	6
	Modify	74	103	177	0	0	0
Total	/	1647	1620	3267	207	26	233

Table 1: Sentence-level AR quality annotation statistics on *elementary* and *college essays*.

ID	Original Draft Sentence	Revised Draft Sentence	Revision Type	Revision Purpose	Quality Label	
1	According to the text. "The people in Sauri have made amazing progress in just eight years."	According to the text, "The people in Sauri have made amazing progress in just eight years."	Modify	Surface	N/A	N/A
2	This tells me that the people of Sauri have made better living arrangements in eight years and it all did pay off.	This tells me that the people of Sauri have made better living arrangements in eight years and it all did pay off, from how they used to live.	Modify	Content	N/A	N/A
3	The people of Sauri did do great progress.	The people of Sauri did do great progress.	N/A	N/A	N/A	N/A
4	...	...	...	...	...	...
5	This lets me know that since there might be a few diseases that might affect anyone at ant time so the hospital has made medicine that can cure those diseases, so they gave that medicine to any one who needed it for free.		Delete	Content	Irrelevant Evidence	Unsuccessful
6		This piece of text lets me know that the hospital, Yala Sub District, has free medicine for diseases that are most common around where they live.	Add	Content	Paraphrase Reasoning	Successful
7		In Sauri people had to pay a fee, which the people of Sauri couldn't afford.	Add	Content	not LCE Reasoning	Unsuccessful
8	...	...	...	...	...	...
9	I can tell that the people of Sauri must of thought that children needed education money not so they didn't ask people for the school fees, and the kids wouldn't go hungry during school hours they served the children lunch.		Delete	Content	LCE Reasoning	Successful
10	...	...	...	...	...	...

Table 2: Example of revision annotations for an *elementary essay*. Note that the successful and unsuccessful labels in the last column are only used for evidence and reasoning content revisions; other purpose types in (Zhang et al., 2017) are not in the scope of this study as we only focus on evidence use and reasoning in argumentative writing.

*mentary essays* written by grade 5 to 6 students who were taking the Response to Text Assessment (Correnti et al., 2013). 296 students wrote an essay in

response to a prompt about the United Nation's Millennium Villages Project (MVP). The students then revised their essays in response to formative feed-

back from an Automatic Writing Evaluation (AWE) system that used rubric-based algorithms to assess the quality of evidence use and reasoning (Zhang et al., 2019; Wang et al., 2020). The other 300 students did the same tasks for an essay prompt about Space Exploration (Space). We combined the collected essays from the two essay prompts because students shared similar argumentative writing skills and the scoring rubric and feedback messages were constant across prompts.

### 3.2 Preprocessing

We preprocessed collected *elementary essays* for annotations. First, sentences from original and revised drafts were aligned into pairs of original sentence (OS) and revised sentence (RS) using a sentence alignment tool Bertalign (Liu and Zhu, 2022). The aligned pairs were programmatically labeled with *no change* if OS and RS are the same, *modifying* if OS and RS are not empty but not same, *adding* if OS is empty but RS not, or *deleting* if RS is empty but OS not. The changed alignments were automatically classified into *surface* and *content* revisions by a pretrained classifier. Note that the sentence alignments and classification were first done by the system and then manually justified and corrected by annotators, and only aligned *content* revisions were used for annotations.

### 3.3 Annotations

We used the Revisions of Evidence use and Reasoning (RER) scheme (Afrin et al., 2020) to annotate revision purposes in *elementary essays*, which encodes the nature of students’ revision of evidence use and reasoning. Evidence use refers to the selection of relevant evidence from a given source article to support a claim, while reasoning means a reasoning process of connecting the evidence to the claim. Thus, the *content* revisions are annotated with claim-related, evidence, and reasoning revisions. The RER scheme only applies to evidence and reasoning, where evidence revisions were labeled with *relevant*, *irrelevant*, *repeated evidence*, *non-text based* and *minimal*, and reasoning revisions were labeled with *linked claim-evidence (LCE)*, *not LCE*, *paraphrase evidence*, *generic*, *commentary*, and *minimal*.

Furthermore, we followed the AR quality scheme (Afrin and Litman, 2023) to programmatically encode annotated RER labels (revision purposes) into *successful* and *unsuccessful* revisions. The *relevant* evidence was encoded as *successful*

	Space Essays		MVP Essays	
	RER#	Kappa	RER#	Kappa
Reasoning	148	0.86	135	0.84
Evidence	108	0.89	136	0.80

Table 3: Annotation agreement for reasoning and evidence RER annotations in a sample of 20 percent of *elementary essays* regarding Space and MVP prompts.

while the *repeated*, *non-text based*, and *minimal* evidence were encoded as *unsuccessful*. The *LCE* and *paraphrase* reasoning were encoded as *successful*. The *not LCE*, *paraphrase evidence*, *generic*, *commentary*, and *minimal* reasoning were encoded as *unsuccessful*. Table 1 shows label distributions in *elementary essays* and *college essays* where *elementary essays* have almost an even number of reasoning and evidence annotations. The *adding* revisions are the most frequent ARs across two essays. Samples of annotations for *elementary essays* and *college essays* are shown in Table 2 and Table 6 (in Appendix A), respectively. In practice, the RER annotations were done by one expert annotator. We sampled about 20 percent of annotated essays about both Space and MVP prompts and asked another well-trained annotator to annotate the sampled essays. The two-annotator Kappa scores are shown in Table 3.

## 4 Methods

### 4.1 Preliminary

In this section, we introduce notations for the AR quality prediction task. We denote  $R_1$  and  $R_2$  as original and revised sentences in the original and revised drafts, respectively. In particular,  $R_1$  is always empty in *adding* ARs (e.g., row #6 in Table 2);  $R_2$  is always empty in *deleting* ARs (e.g., row #9 in Table 2); neither  $R_1$  nor  $R_2$  are empty in *modifying* ARs (e.g., row #1 in Table 2). Thus, we only use  $R_1$  in *deleting* and  $R_2$  in *adding* ARs. In terms of *modifying* ARs, we only use  $R_2$  because  $R_2$  is a revised version of  $R_1$  thus are very close to  $R_1$  (e.g., row #2 in Table 2). In addition, we denote ACs as a couple of sentences related to ARs in their corresponding drafts, where  $C_1$  represents the ACs of  $R_1$  in the original draft and  $C_2$  represents the ACs of  $R_2$  in the revised draft (details in Sec. 4.2), respectively. To this end, we formulate the task of predicting AR quality as classifying the AR-AC pairs  $\{R_i, C_i\}$  into *successful* and *unsuccessful* labels, where  $i = 1, 2$ . Specifically, we use pair

$\{R_1, C_1\}$  for *deleting* and  $\{R_2, C_2\}$  for *adding* and *modifying* ARs.

## 4.2 Argumentative Context

Consistent with long-established models of argumentation such as Toulmin’s model (Toulmin, 1958), well-developed arguments are characterized by the alignment of claim, evidence, and warrants (i.e., reasoning related to why the evidence supports the claim) (Reznitskaya et al., 2008). For example, the appropriateness of a piece of evidence for advancing an argument is context-dependent because that judgment is determined relative to an author’s prior claim(s) or reason(s). As a case in point, the unsuccessful AR #372 shown in Figure 1 would have been unobservable absent an understanding of the author’s claim or argument’s context. Recent work by Afrin and Litman (2023) has used short and long text segments immediately before and after the AR as context for predicting AR quality, however, the study has some significant drawbacks. First, the window size of the contexts is an unpredictable parameter because a reasoning sentence could refer to the evidence far ahead of the AR (e.g., reasoning in row #6 refers to the evidence in row #1 in Table 2). Second, location-based contexts did not explain why ACs make a difference to ARs from an argumentative perspective and thus fail to analyze the argumentative roles of ACs in AR quality predictions. As we noted above, the evaluation of a reasoning sentence as desirable depends on whether it appropriately references evidence or claims in the student’s essay, but this relationship has not been explored in prior revision research. Thus, in the current study, we define three ACs to study their relationship to AR quality: **(1) AC-Claim:** the context containing essay claims or arguments; **(2) AC-Reasoning:** the context containing reasoning related to the claim or evidence in the essay; **(3) AC-Evidence:** the context containing evidence to support or oppose claims.

## 4.3 ChatGPT Prompts

Pretrained ChatGPT on a series of GPT3.5 models has shown promising results in solving information extraction (Li et al., 2023) and summarization (Yang et al., 2023) tasks in zero-shot settings, however, doing the two tasks at the same time has not been explored in generating ACs. Therefore, we developed two versions of ChatGPT prompts that generate useful ACs for predicting AR quality: (1) Single prompts that generate ACs in one pass

and (2) Chain-of-Thought prompts that generate ACs in two passes.

### 4.3.1 Single Prompts

In this section, we introduce Single prompts for AC generations. Basically, we need ChatGPT to generate useful ACs for AR quality predictions by reading the student essays. We limit the generation to a two-sentence length for two reasons. First, the generated ACs will be used in an AR-AC pair  $\{R_i, C_i\}$ , where  $R_i$  is normally one sentence, thus long ACs ( $C_i$ ) paired with short ARs ( $R_i$ ) will make the AR quality prediction model (introduced in Sec. 4.4) learn to attend to the context rather than the revisions. Second, the most intuitive location-based baseline (Base-Short in Sec. 5) uses the adjacent sentences before and after target ARs, which contain at most two sentences. Therefore we limit the generations to exact two sentences, which can be done with a single zero-shot prompt please summarize [X] in the essay [Y] in two sentences, where [X] slot is one of the *claim*, *reasoning*, and *evidence*, [Y] is an input essay.

### 4.3.2 Chain-of-Thought Prompts

In addition to Single prompts, Chain-of-Thought (CoT) prompts (Wei et al., 2022) are conceptually simple yet effective in multiple reasoning tasks. We adopt this idea and use zero-shot-CoT prompts to generate ACs, which run prompting in two passes but do not require step-by-step few-shot examples.

**The first-pass CoT prompt to extract ACs.** The first pass of the CoT prompts is to extract claim, evidence, and reasoning sentences from input essays. We aim to extract exact sentences from input essays without introducing any external knowledge in ChatGPT itself. The first-pass prompt is: please list [X] sentences in the essay [Y], where [X] slot is chosen from one of the *claim*, *reasoning*, and *evidence*, and [Y] is an input essay. The extracted ACs are formulated as a list of sentences from the input essays, where the length of the list ranges from one to dozens because *claim*, *reasoning*, and *evidence* have multiple instances in an essay. To make sure the extracted ACs are informative and not exceeding the word limits of DistilRoBERTa encoders (see Sec. 4.4), we perform summarization on the extracted sentences in the second-pass prompt.

**The second-pass CoT prompt to summarize ACs.** The second prompt is continued to the first

<p><b>Input Essay #104 (First Draft)</b></p> <p>No the author did not convince me that winning the fight against poverty is achievable in our lifetime. I just think that it would be nearly impossible because there are a lot of people dying and they don't have too much money. In the text it states that, " The solutions are simple, yet 20,000 kids are die from the disease each day. So sad and so illogical. Bed nets could save millions of lives." If there are that many people dying a day, then how will the doctors have time to make cures or anything? All of the solutions could take as long as more that 10 years. That would be really hard for all of he doctors to be working so hard, but the solutions to take for ever. In the text it states that , " The goals are supposed to be met by 2025 ;some other targets are met by 2035." By the time it is 2025, a lot of the people will be sick and they can die if they don't get treated fast enough. The passage says there were at lest 5 people in 1 bed, and some people would not get hardly any treatment, and some of the patients could not afford theses things. I think that it would be nearly impossible to do this mission because many people are dying a day and the doctors still need some more time. The doctors might get to save a few people depending on how many there are, but they will not get save many. In conclusion, you may have a to of solutions to your problem, but you will not always have the answers in time that you would want.</p>
<p><b>Generated AC-Claim</b></p> <p><b>Single Prompt:</b> Please summarize claim in the essay in two sentences.</p> <p>[Output] The author fails to convince the writer that winning the fight against poverty is achievable in their lifetime due to the high daily death toll and the lengthy time frame for solutions. The passage highlights the dire state of poverty with overcrowded hospitals and inadequate treatment options, ultimately underscoring the difficulty in finding effective answers.</p> <p><b>CoT Prompt:</b> 1) Please list claim sentences in the essay. 2) Please summarize the listed claim in two sentences.</p> <p>[Output] The author failed to convince the reader that ending poverty is feasible due to the high death rate and long time frame for implementing solutions. The present systems are inadequate, leading to overcrowded hospitals and lack of access to medical care.</p>
<p><b>Generated AC-Reasoning</b></p> <p><b>Single Prompt:</b> Please summarize reasoning in the essay in two sentences.</p> <p>[Output] The author did not persuade the writer that poverty can be eradicated within our lifetime due to the high mortality rate of 20,000 people living in poverty dying every day. The solutions proposed may take too long and inadequate healthcare results in many people not receiving the necessary treatment.</p> <p><b>CoT Prompt:</b> 1) Please list reasoning sentences in the essay. 2) Please summarize the listed reasoning in two sentences.</p> <p>[Output] The author's argument that winning the fight against poverty is unlikely is supported by the fact that 20,000 people die from poverty every day and some proposed solutions will take years to implement. The passage also highlights the dire conditions and lack of access to treatment for many impoverished individuals.</p>
<p><b>Generated AC-Evidence</b></p> <p><b>Single Prompt Output:</b> Please summarize evidence in the essay in two sentences.</p> <p>[Output] The author fails to convince the reader that winning the fight against poverty is achievable in our lifetime due to the high rate of poverty-related deaths and the lengthy timeframe for implementing solutions. The overcrowding in hospitals and lack of access to treatment further supports the argument that successful poverty eradication may not be possible.</p> <p><b>CoT Prompt Output:</b> 1) Please list evidence sentences in the essay. 2) Please summarize the listed evidence in two sentences.</p> <p>[Output] Despite goals set to be met by 2025 and 2035, poverty-fueled deaths continue at a rate of 20,000 per day, with potential solutions taking more than 10 years to implement. As a result, many are left without access to healthcare and overcrowded hospitals, with some unable to afford or receive necessary treatment.</p>

Figure 2: The input and output of the ChatGPT with zero-shot Single and CoT prompts for an *elementary* essay.

prompt, following an extraction-summarization CoT. The prompt is `please summarize [X] in two sentences, where [X] slot is chosen from the claim, reasoning, and evidence sentences extracted in the first prompting pass, which ensures the outputs in a length of exact two sentences.` Figure 2 and Figure 5 (in Appendix A) show examples of the zero-shot Single and CoT prompts that help ChatGPT generate ACs in *elementary* and *college* essays, respectively.

#### 4.4 AR Quality Prediction

We define AR quality prediction as a binary classification of AR-AC pair  $\{R_i, C_i\}$  (see Sec. 4.1), where  $R_i$  is annotated and  $C_i$  is generated by ChatGPT. Prior works employed BERT-BiLSTM architecture to train revision classifiers (Antonio and Roth, 2020; Afrin and Litman, 2023). Instead,

we use DistilRoBERTa (Sanh et al., 2019) as text encoders for both annotated ARs and ChatGPT-generated ACs. The last hidden layers of the DistilRoBERTa encoders are fed to an average-pooling layer and then connected to a multi-layer perceptron classifier that contains a sequence of batch normalization layer, ReLU layer, dropout layer with a 0.5 rate, and Sigmoid layer. The overall framework is shown in Figure 3.

## 5 Experiments

To answer the RQs, we implemented location-based contexts as baselines, and a series of ACs as comparable methods:

- **Base-Short:** We implement a standard revision prediction baseline that uses the adjacent sentences immediately before and after a revision as



Contexts	Prompts	Reasoning & Evidence ARs			Reasoning ARs			Evidence ARs		
		Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Base-Short	N/A	67.79	67.17	67.29	70.29	70.01	69.94	63.42	62.60	62.67
Base-Long	N/A	68.45	68.01	68.06	69.99	69.76	69.71	65.38	64.69	64.71
AC-Claim	Single	<u>70.31</u>	<u>69.83</u>	<u>69.91</u>	<u>72.63</u>	<u>72.47</u>	<u>72.38</u>	66.60	65.33	65.57
AC-Reasoning	Single	<u>70.15</u>	<u>69.67</u>	<u>69.74</u>	72.10	71.95	71.83	66.57	65.60	65.79
AC-Evidence	Single	<u>70.28</u>	<u>69.97</u>	<u>69.93</u>	72.46	72.31	72.13	66.55	65.87	65.85
AC-Claim	CoT	<u>70.09</u>	69.64	69.74	71.83	71.76	71.70	66.74*	65.71*	65.88*
AC-Reasoning	CoT	<b>71.14*</b>	<b>70.81*</b>	<b>70.81*</b>	<b>72.86*</b>	<b>72.80*</b>	<b>72.63*</b>	<b>68.00*</b>	<b>67.00*</b>	<b>67.16*</b>
AC-Evidence	CoT	<u>70.43*</u>	<u>70.03*</u>	<u>70.01*</u>	<u>72.48*</u>	<u>72.34*</u>	<u>72.20*</u>	66.76*	66.06*	66.05*

Table 4: Experimental results on *elementary essay* corpus. The bold numbers are the best results. The underlined numbers statistically outperformed the strong (Base-Long) baseline in a paired t-test with  $p < 0.05$ . The asterisks indicate zero-shot-CoT prompts are better than zero-shot-Single prompts.

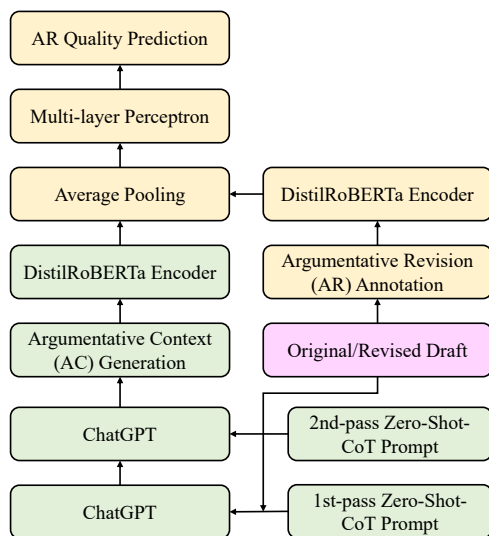


Figure 3: The overall framework of AR quality predictions, where the pink box is input; the green boxes are our proposed; the yellow boxes are existing methods.

contexts (Afrin and Litman, 2023).

- **Base-Long:** We implement a strong baseline that considers all the sentences that are revised around a target revision until an unchanged sentence is found (Afrin and Litman, 2023).
- **AC-Claim:** We use AC-Claim as the contexts that are generated by zero-shot Single and CoT prompts, respectively.
- **AC-Reasoning:** We use AC-Reasoning as contexts. The two versions use zero-shot Single and CoT prompts, respectively.
- **AC-Evidence:** We use AC-Evidence as contexts. The two versions are generated by zero-shot Single and CoT prompts, respectively.

In the implementation, we built the framework pipeline with PyTorch<sup>3</sup> and generated two versions

<sup>3</sup><https://pytorch.org>

Contexts	Prompts	Precision	Recall	F1
Base-Short	N/A	59.95	60.06	58.61
Base-Long	N/A	61.21	61.60	59.48
AC-Claim	Single	63.50	64.08	62.27
AC-Reasoning	Single	63.06	63.62	61.86
AC-Evidence	Single	<u>65.76</u>	66.40	<u>64.71</u>
AC-Claim	CoT	64.01*	64.96*	62.93*
AC-Reasoning	CoT	63.84*	64.33*	62.74*
AC-Evidence	CoT	<b>68.20*</b>	<b>68.05*</b>	<b>66.32*</b>

Table 5: Experimental results on reasoning ARs in *college essays*. The bold numbers are the best results. The underlined numbers statistically outperformed the strong (Base-Long) baseline in a paired t-test with  $p < 0.05$ . The asterisks indicate zero-shot-CoT prompts are better than zero-shot-Single prompts.

of ACs using ChatGPT3.5-turbo API<sup>4</sup>. We used pretrained DistilRoBERTa-Base from Huggingface<sup>5</sup> as text encoders, and optimized cross-entropy loss with Adam optimizer on a GeForce RTX 3090 GPU. We set the batch size as 16 and the learning rate as  $5e-5$  with 5% decays every 4 epochs. We conducted 10-fold cross-validation, where 80% of each 9-fold set was used for training, 20% for parameter tuning, and the rest 1-fold set for testing. Finally, we ran the ChatGPT generation and the experiment pipeline three times and reported 3-seed-average macro Precision, Recall, and F1 on all the test sets. The implementation code is available at <https://github.com/ZhexiongLiu/Revision-Quality-Prediction>.

## 6 Results and Discussion

Table 4 shows the experimental results for different sets of revisions from the *elementary essay* corpus: all reasoning and evidence revisions, just

<sup>4</sup><https://platform.openai.com>

<sup>5</sup><https://huggingface.co>

reasoning revisions, and just evidence revisions, respectively. We observed that both the proposed Single and CoT versions of ACs outperformed both baselines, with many of the CoT ACs significantly better than the strong (Base-Long) baseline. This answered **RQ1** that ACs can help AR quality predictions. In reasoning ARs, excellent performance was yielded in using AC-Claim, AC-Reasoning, and AC-Evidence. This is because reasoning ARs might need claims to verify their usefulness and incorporate evidence and reasoning to check their relevance. Moreover, evidence ARs achieved the best with AC-Reasoning, which makes sense that identifying evidence AR requires related reasoning contexts that have information linking the evidence. Another interesting finding is that the Base-Long performed better than the Base-Short in evidence ARs but worse in reasoning ARs. This suggests that the longer context is not always helpful in the case that evidence contexts are usually sparsely distributed in the essay so the longer context will introduce more noise. It also suggests that reasoning sentences are mostly adjacent to other reasoning contexts and can be well captured by neighboring sentences. Furthermore, the observation that reasoning ARs results are generally better than evidence ARs indicates that reasoning ARs might be self-justifiable which means it might require fewer contexts than the evidence to identify AR quality. These observations answered **RQ2** that reasoning contexts are mostly useful, and both reasoning, claim, and evidence contexts benefit AR quality predictions. In addition, CoT prompts are generally better than Single prompts in most reasoning and evidence ARs, which indicates that identifying AR quality requires some contexts that might not be generated with Single prompts. This answered **RQ3** that CoT prompts are generally better than Single prompts.

We also evaluated the effectiveness of ACs on the *college essay* benchmark. Note that [Afrin and Litman \(2023\)](#) conducted data augmentation with a simple synonym replacement because they argued that it was impossible to obtain reasonable results without training on augmented data. We hypothesized that data augmentation will introduce noise but the limited data can yield reasonable results training with the DistilRoBERTa-based model. Therefore, we did not do data augmentation and compared AC-based methods to our implemented standard and strong baselines on reasoning revisions

#### Revision #372: AC-Claim

<claim> The essay highlights progress made in Sauri, including free medicine and bed net provision, as well as the positive impact of **providing lunch to children, to argue that poverty can be reduced...** </claim>

#### Revision #592: AC-Claim

<claim> The essay argues that poverty and lack of resources can be tackled in our lifetime with examples such as bed nets to prevent malaria, updated hospitals to prevent the spread of diseases and access to education... </claim>

Figure 4: Two pieces of ChatGPT-generated AC-Claims. The red bold is the context to identify Revision #372 is a *already existed* adding, while #592 is a *relevant* adding toward their contexts in Figure 1.

sions (excluding the rare evidence revisions as shown in Table 1). Results in Table 5 show that the DistilRoBERTa model is able to learn from even small-size data without data augmentation. In addition, AC-based methods perform better than both the standard and strong baselines, where AC-Evidence has significant improvement. This again suggests that ACs are generally useful for predicting AR quality and CoT prompts are generally better than Single prompts for generating useful ACs. Moreover, we observed that AC-Evidence generated by Single and CoT prompts is better than the other ACs. It is slightly different from the reasoning column in Table 4. This might suggest that revisions in *college essays* may focus on evidence revisions that match generated evidence ACs. However, *claim* and *reasoning* results have similar F1 scores across two versions of prompts, which might suggest the extracted AC-Claim and AC-Reasoning are similar in *college essays* (e.g., prompting outputs in Figure 5 in Appendix A), which might be because *college essays* have claim and reasoning sentences disentangled. In general, CoT prompts are somewhat better than Single prompts in AC-Claim and AC-Reasoning generation, and both Single and CoT prompts are promising in AC-Evidence generation.

As a case study, we examine the effectiveness of ACs in Revision #372 and #592 presented in Figure 1. The ChatGPT-generated AC-Claim is shown in Figure 4, where the red bold sentence “*providing lunch to children, to argue that poverty can be reduced*” is helpful to identify that the added sentence, “*It was hard for them to concentrate though,*

as there was no midday meal.” in Revision #372 is a *already existed* evidence, and thus it was an *unsuccessful* revision. However, AC-Claim in Revision #592 does not show repeated but *relevant* information, and thus the AR is regarded as *successful*.

## 7 Conclusion

This work studies the relationship between Argumentative Contexts (ACs) and Argumentative Revisions (ARs) in argumentative writing. In particular, we use zero-shot-CoT prompts to facilitate ChatGPT-generated ACs for AR quality predictions. The experiments on our *elementary essays* corpus and publicly available *college essays* benchmark demonstrate the superiority of the proposed ACs over existing location-based context baselines, which proposes a new direction for predicting AR quality. The analysis suggests that most evidence ARs need reasoning ACs, and reasoning ARs need a diverse set of claims, evidence, and reasoning ACs to predict their quality.

## 8 Limitations

Our experiments were built on perfect sentence alignments in the original and revised essay drafts, thus the performance could be lower in the real end-to-end Automated Writing Evaluation (AWE) system. In addition, our corpus is small due to expensive annotation processes, which makes it challenging to train or finetune large language models. Also, we only focus on revisions in argumentative writing, specifically, we focus on the evidence and reasoning revisions, however other revisions like claim revisions are not used. Furthermore, the revised drafts were done after providing feedback on the original drafts, which means the revised student essays are likely to follow the instructions in the feedback but we did not use this information for revision quality predictions, which will be used in our future work.

Our proposed Argumentative Contexts (ACs) are generated by ChatGPT which is not free for the whole community. Also, ChatGPT-generated ACs have small randomness, which is also the reason we did 3-seed runs in the experiments. In addition, the ACs are essay-level context which means different revisions in the same essay use the same context. It could be tailored to have sentence-level ACs where each sentence-level revision has slightly different revision purposes, but it would cost more

time and money. Moreover, our proposed zero-shot-CoT prompts perform better than Single prompts by small margins in specific cases, which indicates that Chat-GPT is limited to conducting CoT extraction and summarization to handle complex wording and sentence structure. Therefore, we might need to redesign the prompts in our future work.

## 9 Ethics

Our corpus was collected under standard protocols that were approved by an institutional review board. Our annotated data is not publicly available which ensures the safety of private information of the students, and thus will not pose any ethical concerns because other researchers can not access our data and replicate our results. Our future work is to incorporate proposed methods in real AWE systems to evaluate student writings and provide informative feedback based on predictions. But there is a risk that the system might give poor advice based on incorrect AR quality predictions, given that the model may learn biases with small annotated data.

## Acknowledgments

The research was supported by the National Science Foundation Award #2202347 and a gift from CloudBank. The opinions expressed were those of the authors and did not represent the views of the institutes. We would like to thank anonymous reviewers and Pitt PETAL group for their valuable feedback on this work.

## References

- Tazin Afrin and Diane Litman. 2023. [Predicting desirable revisions of evidence and reasoning in argumentative writing](#). In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2550–2561, Dubrovnik, Croatia. Association for Computational Linguistics.
- Tazin Afrin, Elaine Lin Wang, Diane Litman, Lindsay Clare Matsumura, and Richard Correnti. 2020. Annotation and classification of evidence and reasoning revisions in argumentative writing. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, Seattle, Washington, USA (Remote).
- Talita Anthonio and Michael Roth. 2020. [What can we learn from noun substitutions in revision histories?](#) In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1359–1370, Barcelona, Spain (Online). International Committee on Computational Linguistics.

- Beata Beigman Klebanov and Nitin Madnani. 2020. [Automated evaluation of writing – 50 years and counting](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7796–7810, Online. Association for Computational Linguistics.
- 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](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayanan Pilla, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. [Palm: Scaling language modeling with pathways](#).
- Richard Correnti, Lindsay Clare Matsumura, Laura Hamilton, and Elaine Wang. 2013. Assessing students’ skills at writing analytically in response to texts. *The Elementary School Journal*, 114(2):142–177.
- 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.
- Takumi Ito, Tatsuki Kuribayashi, Hayato Kobayashi, Ana Brassard, Masato Hagiwara, Jun Suzuki, and Kentaro Inui. 2019. [Diamonds in the rough: Generating fluent sentences from early-stage drafts for academic writing assistance](#). In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 40–53, Tokyo, Japan. Association for Computational Linguistics.
- Omid Kashefi, Tazin Afrin, Meghan Dale, Christopher Olshefski, Amanda Godley, Diane Litman, and Rebecca Hwa. 2022. [Argrewrite v.2: an annotated argumentative revisions corpus](#). *Language Resources and Evaluation*, pages 1574–0218.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. [Large language models are zero-shot reasoners](#). In *NeurIPS*.
- Bo Li, Gexiang Fang, Yang Yang, Quansen Wang, Wei Ye, Wen Zhao, and Shikun Zhang. 2023. Evaluating chatgpt’s information extraction capabilities: An assessment of performance, explainability, calibration, and faithfulness. *ArXiv*, abs/2304.11633.
- Lei Liu and Min Zhu. 2022. Bertalign: Improved word embedding-based sentence alignment for chinese–english parallel corpora of literary texts. *Digital Scholarship in the Humanities*.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.
- Wenchuan Mu and Kwan Hui Lim. 2022. [Revision for concision: A constrained paraphrase generation task](#). In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 57–76, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.
- Jan Wira Gotama Putra, Simone Teufel, and Takenobu Tokunaga. 2021. Parsing argumentative structure in english-as-foreign-language essays. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 97–109.
- Alina Reznitskaya, Richard C Anderson, Ting Dong, Yuan Li, Il-Hee Kim, and So-Young Kim. 2008. *Learning to think well: Application of argument schema theory to literacy instruction*. The Guilford Press.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.
- Antonette Shibani, Simon Knight, and Simon Buckingham Shum. 2018. Understanding revisions in student writing through revision graphs. In *International Conference on Artificial Intelligence in Education*, pages 332–336, Cham. Springer International Publishing.

- Gabriella Skitalinskaya, Jonas Klaff, and Henning Wachsmuth. 2021. [Learning from revisions: Quality assessment of claims in argumentation at scale](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1718–1729, Online. Association for Computational Linguistics.
- Wei Song, Ziyao Song, Ruiji Fu, Lizhen Liu, Miaomiao Cheng, and Ting Liu. 2020. Discourse self-attention for discourse element identification in argumentative student essays. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2820–2830.
- Christian Stab and Iryna Gurevych. 2014. Identifying argumentative discourse structures in persuasive essays. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 46–56.
- Stephen Edelston Toulmin. 1958. *The Uses of Argument*. Cambridge university press.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Boshi Wang, Xiang Deng, and Huan Sun. 2022. Iteratively prompt pre-trained language models for chain of thought. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2714–2730.
- Elaine Lin Wang, Lindsay Clare Matsumura, Richard Correnti, Diane Litman, Haoran Zhang, Emily Howe, Ahmed Magooda, and Rafael Quintana. 2020. *erevis(ing): Students’ revision of text evidence use in an automated writing evaluation system*. *Assessing Writing*, 44:100449.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. [Chain of thought prompting elicits reasoning in large language models](#). In *Thirty-sixth Conference on Neural Information Processing Systems (NeurIPS 2022)*.
- The Writing Mentor. 2016. ETS writing mentor, <https://mentormywriting.org/>, [online; accessed 02-06-2019].
- Diyi Yang, Aaron Halfaker, Robert E. Kraut, and Edward H. Hovy. 2017. Identifying semantic edit intentions from revisions in wikipedia. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP’17*, pages 9–11, Copenhagen, Denmark. Association for Computational Linguistics.
- Xianjun Yang, Yan Li, Xinlu Zhang, Haifeng Chen, and Wei Cheng. 2023. Exploring the limits of chatgpt for query or aspect-based text summarization. *ArXiv*, abs/2302.08081.
- Fan Zhang, Homa B Hashemi, Rebecca Hwa, and Diane Litman. 2017. A corpus of annotated revisions for studying argumentative writing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1568–1578.
- Fan Zhang, Rebecca Hwa, Diane Litman, and Homa B Hashemi. 2016. Argrewrite: A web-based revision assistant for argumentative writings. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: Demonstrations*, pages 37–41.
- Fan Zhang and Diane Litman. 2015. Annotation and classification of argumentative writing revisions. In *Proceedings of the 10th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 133–143, Denver, Colorado. Association for Computational Linguistics.
- Fan Zhang and Diane Litman. 2016. Using context to predict the purpose of argumentative writing revisions. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1424–1430.
- Haoran Zhang, Ahmed Magooda, Diane Litman, Richard Correnti, Elaine Wang, LC Matsumura, Emily Howe, and Rafael Quintana. 2019. *erevise: Using natural language processing to provide formative feedback on text evidence usage in student writing*. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9619–9625.

## A Appendix

ID	Original Draft Sentence	Revised Draft Sentence	Revision Type	Revision Purpose	Quality Label	
1	A mother who would have no other way of reaching her children can easily speak to them or leave a message via voicemail.	A mother who would have no other way of reaching her children can easily speak to them or leave a message via voicemail.	N/A	N/A	N/A	
2	Technology makes it possible to reach anyone at any time.		Delete	Content	LCE Reasoning	Unsuccessful
3		In addition, technology makes it possible to increase the amount of communication between people drastically.	Add	Content	LCE Reasoning	Successful
4	...	...	...	...	...	...
5	People from different continents who may have never met before can now have conversations every day; even those from a remote location are available to the world, provided they have the Internet.	People from different continents who may have never met before can now have conversations every day; even those from a remote location are available to the world, provided they have both the Internet and a corresponding device.	Modify	Surface	N/A	N/A
6		How could a cold inanimate screen replace seeing the emotions and expressions of a loved one?	Add	Content	not LCE Reasoning	Unsuccessful
7		An essential thing to consider is that while perhaps it may be harder to convey one's full message complete with feelings through the Internet, the fact remains that in a changing world where people are busier and farther away, electronic devices are helping everyone keep in contact with each other at any time of the day and at any location.	Add	Content	LCE Reasoning	Successful
8		Those who argue for the retardation of technology simply cannot accept that the world is developing to be more tech driven; as more and more people have electronic devices, they are also changing to be more used to this kind of communication.	Add	Content	not LCE Reasoning	Unsuccessful
9	...	...	...	...	...	...

Table 6: Example of revision annotations for a *college essay*.

**Input Essay #107 (First Draft)**

Throughout history, society has undergone advances in many realms of life. These realms include politics, social issues, education, and also technology, which is one of the most notable. The proliferation of technology has included the creation of computers, which can be used to research new information in a short period of time. Technology also allows people to send and receive information, using devices such as cellular phones. Because of this ease of information transfer, technology has been able to enrich interpersonal relationships by providing another avenue for communication. Technology has enriched interpersonal relationships by allowing communication between people when they are not in physical vicinity. For example, students in college may develop friendships during the school year, but during the summer they may be unable to keep in physical contact with these friends depending on how far they live from one another, as many students go home for the summer. Because verbal communication is an important part of maintaining relationships, this distance could pose a threat to the maintenance of these relationships. However, staying in contact via texts, calls, and social media gives students the opportunity to remain in contact, even if the distance between them is large. This communication lets all parties know that even while not in the vicinity of one another, they are still able to enjoy each other and have similar interests. Because of this, the relationship has the potential to be even stronger upon their return back to school. Technology also enriches interpersonal relationships by increasing the frequency with which people can communicate. Due to technology, people can communicate several times throughout the day because of the speed of information transfer. For Example, if someone has an issue that they would like to relay to another person, they are able to notify them almost instantly due to the speed of communication. Additionally, they are able to discuss the matter back and forth. This can lead to greater conflict resolution as it allows the issue to be resolved faster, rather than waiting until the next time the parties involved are in contact with each other. The removal of this time-lapse between contact is beneficial as it could eliminate the emergence of more negative emotions. In sum, contact without physical vicinity, as well as, increased frequency of communication between people displays how the proliferation of technology has been able to increase the strength of interpersonal relationships.

**Generated AC-Claim**

**Single Prompt:** Please summarize claim in the essay in two sentences.

[Output] The proliferation of technology has enriched interpersonal relationships by providing another avenue for communication. This is due to the ease of information transfer and increased frequency of communication through devices like cellular phones, which allow people to stay in touch even when physically apart.

**CoT Prompt:** 1) Please list claim sentences in the essay. 2) Please summarize the listed claim in two sentences.

[Output] Technology has improved interpersonal relationships by providing more ways to stay in touch over long distances. Frequent communication through various technologies increases the bond between individuals.

**Generated AC-Reasoning**

**Single Prompt:** Please summarize reasoning in the essay in two sentences.

[Output] Technology has enriched interpersonal relationships by allowing communication between people when they are not physically together. it increases the frequency and speed of communication, which can lead to conflict resolution and the strengthening of relationships.

**CoT Prompt:** 1) Please list reasoning sentences in the essay. 2) Please summarize the listed reasoning in two sentences.

[Output] The use of technology has led to advancements in politics, social issues, education, and communication. technology has enriched interpersonal relationships by allowing for frequent and efficient communication, which has led to stronger relationships with fewer negative emotions.

**Generated AC-Evidence**

**Single Prompt Output:** Please summarize evidence in the essay in two sentences.

[Output] Advancements in technology have allowed for easy information transfer and communication through devices like computers and phones, enriching interpersonal relationships. communication without physical vicinity and increased frequency of contact have increased the strength of relationships, aiding conflict resolution and promoting positive emotions.

**CoT Prompt Output:** 1) Please list evidence sentences in the essay. 2) Please summarize the listed evidence in two sentences.

[Output] Technology has led to advancements in areas like politics, education, communication and social issues. it allows for quicker research and access to new information, easier communication and interpersonal relationships.

Figure 5: The input and output of the ChatGPT with zero-shot Single and CoT prompts for a *college essay*.

# Reconciling Adaptivity and Task Orientation in the Student Dashboard of an Intelligent Language Tutoring System

Leona Colling and Tanja Heck and Detmar Meurers

Department of Computational Linguistics

University of Tübingen

Germany

{leona.colling,tanja.heck,detmar.meurers}@uni-tuebingen.de

## Abstract

In intelligent language tutoring systems, student dashboards should display the learning progress and performance and support the navigation through the learning content. Designing an interface that transparently offers information on students' learning in relation to specific learning targets while linking to the overarching functional goal, that motivates and organizes the practice in current foreign language teaching, is challenging. This becomes even more difficult in systems that adaptively expose students to different learning material and individualize system interactions. If such a system is used in an ecologically valid setting of blended learning, this generates additional requirements to incorporate the needs of students and teachers for control and customizability.

We present the conceptual design of a student dashboard for a task-based, user-adaptive intelligent language tutoring system intended for use in real-life English classes in secondary schools. We highlight the key challenges and spell out open questions for future research.

## 1 Introduction

Language learning is a complex, multidimensional process. It is therefore desirable to provide scaffolding support to learners during practice. Intelligent Tutoring Systems (ITS) can implement means for this purpose in an adaptive way and provide students with insights on their progress and performance (Phobun and Vicheanpanya, 2010).

ITS can accommodate individual differences through macro-adaptive exercise selection and provide micro-adaptive support while working on a selected exercise (Slavuj et al., 2017). Macro-adaptive systems therefore automatically determine the order in which learning content is presented, usually based on a static domain model by matching it to learner characteristics such as proficiency and learning styles (Hafidi and Bensebaa, 2014). Each student receives different learning material

which they process at their own pace. The number of exercises a student practices is initially unknown and estimated dynamically after each exercise based on ad-hoc calculations of the student's mastery of the learning object (Rus et al., 2014). Micro-adaptivity, on the other hand, implies that there is no static learning content. Instead, the exercise contents such as hints are dynamically adjusted in order to gradually and individually guide each student towards the correct answers (Lim et al., 2023). Thus, adaptivity improves learning outcomes by adapting to the students' individual needs (Phobun and Vicheanpanya, 2010). Most implementations assign profiles to learners which they generate from training data. Fully adaptive systems then take over all decisions, including structuring and adjusting the learning material based on the learner's profile. This can, however, inhibit them from developing their own learning strategies (Howell et al., 2018). Enabling students to actively engage in the learning decision making process is important to facilitate self-regulation and thus can foster motivation and improve learning outcomes (Lim et al., 2023). Self-regulation can be understood as the students' ability to organize and monitor their own learning behavior and goals by actively managing and shaping their learning environment, such as selecting the next practice target (Schunk and Zimmerman, 2013).

For users to make informed decisions, it is important to show them their personal learning state, according to their interactions with the learning material. Student dashboards generally aim to display information relevant to the student in order to allow them to observe and regulate their learning process. In addition, they provide means to navigate through the learning content (Bull and Kay, 2010). Navigational support is especially relevant and feasible in adaptive systems that incorporate systematically generated, highly variable exercises, such as those following the implementation by Heck and



Meurers (2022). Both information presentation and navigational structure should be provided in a comprehensible and accessible way. This is particularly important when tailoring a system towards young learners, who are still developing their graphical literacy skills, the ability to understand information presented in graphical form (Roberts and Brugar, 2017).

In order to embed individualized adaptive practice with an ITS into real life, task-oriented language learning classrooms put an important additional focus of the student dashboard on linking practice exercises to their overarching functional goal (Andersen, 2019) and integrating with the curriculum the students follow (Phillips et al., 2020).

In systems used for blended learning in school settings, student dashboards must navigate through the content in a way that aligns with the curriculum, through the systems' default sequence or a sequence defined by the teacher, while maintaining enough flexibility to adjust to students' learning preferences.

In addition, teachers need control over certain aspects of the learning material to satisfy the needs of teacher-guided instruction and successfully combine with the classroom-based teaching (Burstein et al., 2012). Controlling the practiced exercises to a certain extent enables them to refer to the material seen by all of their students in subsequent classroom sessions (Feng et al., 2014). Teachers also want to be able to assign deadlines by which students need to complete practice of certain topics (Hertz, 1992).

Since a curriculum-aligned, structured view of the entire learning content conflicts with the adaptive, dynamic content tailored to the student, it is not straightforward to combine both in a single system. We present an approach to address this challenge by supporting multiple navigational strategies and proposing metrics to display progress and performance overviews which take into account the issues faced by traditional metrics with respect to the demands imposed by adaptivity. Specially tailored towards foreign language learning, our dashboard is co-designed with teachers to keep real life implications in mind and support educational practices when integrated into an Intelligent Language Tutoring System (ILTS) for the use in English classes of secondary schools in Germany.

## 2 Related work

An increasing number of ITS integrate student dashboards in form of Open Learner Models (OLM) to expose the users to their learning statistics gathered by the system (Bull et al., 2016). This approach has mainly been applied to higher education (Schwendimann et al., 2017), thus not focusing on the particular requirements of systems used in blended learning settings of secondary school teaching. A noticeable exception constitutes the implementation by Rudzewitz et al. (2019) which, however, does not incorporate a task-oriented embedding of the learning content and lacks sufficient simplicity of the visualizations necessary to guide young learners in their self-regulated learning process.

Since most schools nowadays use task-based teaching approaches for language learning (Andersen, 2019), it is necessary to further adapt student dashboards and OLMs to this concept. In order to represent student progress for the various skills practiced in preparation for the functional target task (Ellis, 2016; Mislevy et al., 2002), the dashboard needs to make these task-essential skills explicit to students. Criterion-referenced feedback, which measures performance against predefined criteria, has been successfully explored and evaluated to this purpose (Mirmakhmudova, 2021; Alawar and Abu-Naser, 2017) and later been integrated into an existing ITS by Colling et al. (2022). Their implementation is tailored towards secondary school children by making the visualizations more accessible for the target age group and incorporating task orientation into the dashboard. To this avail, they highlight the functional goal and group exercises and their performance metrics based on curricular units. This contrasts OLMs, which consider the learning domain as a whole (Bull and Kay, 2010). However, their system is not user-adaptive apart from providing scaffolding feedback so that the student dashboard does not consider the requirements introduced by adaptivity.

Integration of macro-adaptive features into a student dashboard depends on the macro-adaptive strategy the system implements. Knowledge Tracing (KT) approaches keep detailed learner models representing the students' progress for various skills within the practiced domain (Liu et al., 2021) and therefore have the benefit of providing progress metrics for the skills which can be made transparent to students in the form of progress bars

(Effenberger, 2018). They do, however, require large amounts of exercises completed by students to train the underlying model (Chen et al., 2018). Since training data for our target group is not readily available, we cannot reliably determine precise progress values. Other approaches use fixed lengths for exercise sequences with incorrect exercises repeated at the end and merely adapt the required complexity of the exercises to select (e.g., Musa and Mohamad, 2017). The progress bar is then only updated when an exercise is solved correctly. While all macro-adaptive systems adaptively determine exercise sequences within a learning object, they pursue varied strategies to determine the order of learning objects. Depending on the degree of self-regulation a system incorporates, it either (a) dictates the entire learning path for the topic to be practiced (Brusilovsky, 1992), (b) requires the learner to choose the next learning object themselves (Twigg, 2003), or (c) provides navigation support without directly enforcing any specific order (Brusilovsky, 2000). As micro-adaptivity changes the exercise content dynamically while students work on it, assigning fixed complexity scores to exercises becomes unreasonable. Macro-adaptive systems therefore typically do not focus on micro-adaptive strategies, apart from providing scaffolded feedback on all exercises.

The body of research on student-facing progress and performance visualizations applicable in adaptive ITS is growing (e.g., Xia et al., 2019; Loboda et al., 2014; Bull and Kay, 2007). Yet, most of these target higher education and thus do not consider the particular needs of teachers and students in schools. Notable exceptions can be found in the domain of mathematics education (e.g., Long and Alevan, 2017). However, to the best of our knowledge, research on student dashboards in adaptive ILTS especially focusing on the demands and needs of ecologically valid K-12 second language learning classrooms is lacking.

With our user-centered design we want to address this gap and offer an approach for a task-oriented student dashboard supporting different navigational strategies in a system simultaneously implementing macro- and micro-adaptivity and used for secondary school English teaching.

### 3 Dashboard design

Our student dashboard, illustrated in Figure 1, extends a task-oriented dashboard view so that it can

be used in an adaptive ILTS supporting teachers in a blended learning context. Where task orientation and adaptivity requirements clash, special considerations are required. The implementation is based on the assumption that most students use the system on a tablet device in landscape mode. This assumption is backed by observations from real life classrooms. The new dashboard features have been co-designed with English teacher practitioners to ensure initial validity. Following the first three stages of the LATUX workflow (Martinez-Maldonado et al., 2015), based on a needs analysis and iterative interviews with teachers using a low-fidelity prototype, we identified requirements on adaptive ILTS used in blended learning with seventh graders and created a high-fidelity prototype with mock learner data for the proposed learner dashboard. The resulting dashboard is described in the following.

**Structure** The dashboard (see Figure 1) depicts learning content represented in learning units. In accordance with task-based language teaching (Van den Branden, 2016), each unit contains multiple learning targets for grammar or vocabulary practice (e.g., *Simple Past*) which the students need to acquire in order to successfully complete the final communicative target task and its functional goal. In our system, teachers can self-assemble these learning targets into learning units to align with a curriculum, thus supporting different textbooks. Additionally, teachers can define and describe the communicative goal and target task of each learning unit (e.g., *Storytelling: Write a Story! Start with events in the past, describe the present, and then look into the future.*), which will be presented in the dashboard header. Making this link transparent for the students in this way strengthens the connection to the functional target and purpose of practice. Within each learning target of a learning unit, a range of pedagogically motivated realizations of the learning target are listed. *Yes/no questions*, for instance, constitute a realization of the learning target *simple past*. The realizations represent the task-essential language. The system inherent domain model maintains a static, pedagogically motivated order of the learning targets, as well as of the realizations within each target, that have been manually determined by an expert teacher. This structure of the content into coarse and fine-grained content containers makes intermediate acquisition goals visible at different levels

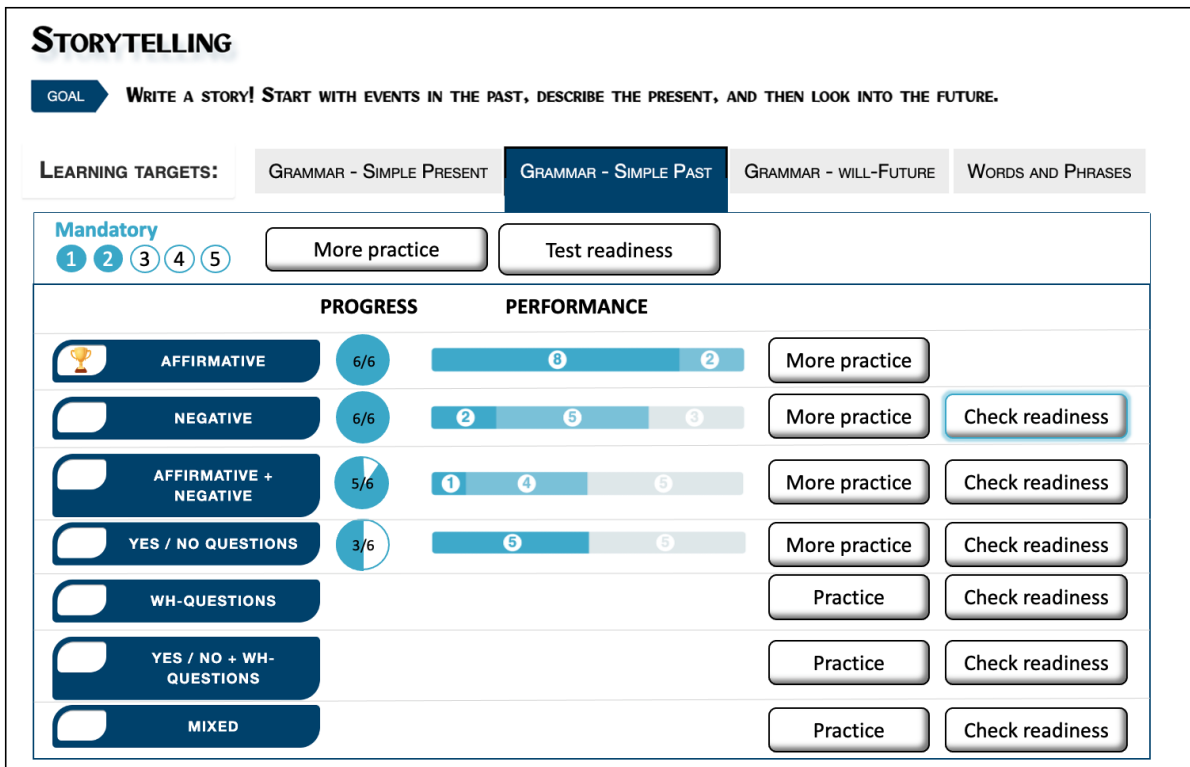


Figure 1: Student dashboard for a user-adaptive, task-oriented ILTS

towards the overarching functional target, which is present at all times, and thus incorporates task-orientation into a student dashboard.

The domain model in our system is based on the curriculum of seventh grade German academic track schools and currently contains 14 grammatical targets, mainly focusing on tenses (i.e., Simple Present, Simple Past, Present Perfect, Past Progressive, Will-Future, Going-to-Future), and their pairwise comparisons, as well as conditional clauses, relative clauses, and comparative forms.

**Navigation** Traditional systems expect high self-regulation from students by requiring them to themselves navigate through the practice material (Sun et al., 2023). Especially for weaker students, these decisions surpass their abilities so that they do better with adaptive systems (Vandewaetere and Clarebout, 2010). In order to support heterogeneous classrooms with both strong and weak students with different navigational preferences, we integrate a hybrid approach, providing options for less and more learner control over the learning content to practice. In the adaptive practice phase, neither students nor teachers choose distinct exercises, this is done by the adaptive exercise sequencing algorithm. The scope of adaptivity varies

depending on the entry point a student chooses. Highly self-regulated students may navigate more autonomously and by themselves select a realization for which the algorithm adaptively sequences exercises. Less self-regulated students, on the other hand, may let the system globally choose both the learning target realization and the exercise.

Although the order displayed via the interface reflects the static order of the pedagogically motivated domain model, the fully adaptive sequence may skip practice of certain realizations for strong students if that realization is also practiced together with other realizations in the same learning target. In the example given in Figure 1, this could for instance be the case for *negative* statements which are also practiced in *affirmative + negative*. Whether a student belongs to the group of strong students for whom realizations are skipped, is based on the student's language proficiency level, which is determined by C-tests periodically administered via the system.

**Progress and performance metrics** A student dashboard serves not only to navigate to the next exercise but also to visualize the student's progress and performance. Given the lack of sufficient training data for our target domain, we cannot use

KT model-based **progress** representations as commonly used in adaptive systems. In traditional systems, progress can be as simple as displaying the ratio of completed exercises out of all exercises (Duan et al., 2010). This is not suitable for macro-adaptive systems, where the number of completed exercises can easily be determined, yet the number of all exercises is unknown before the student has achieved mastery. Bull and McEvoy (2003) suggest an alternative approach which displays the numbers of successfully and unsuccessfully acquired concepts.

We build our progress metric on this idea, but only display successfully acquired linguistic properties for each realization in a pie chart in order to further increase simplicity. Linguistic properties such as *regular verb forms* are defined at a fine-grained linguistic level. Exercises are linguistically analyzed with the annotation pipeline introduced in Rudzewitz et al. (2018) using the Unstructured Information Management Architecture (UIMA, (Ferrucci and Lally, 2004)) and standard natural language processing (NLP) tools, i.e., segmentation, part-of-speech tagging and dependency parsing with ClearNLP (Choi and Palmer, 2012), lemmatization with Morpha (Minnen et al., 2001) and morphological analysis with the Sfst tool (Schmid, 2005). Based on these basic linguistic analyses of the exercise content, including the target answer and any linguistic co-text such as prompts but excluding exercise instructions, the exercise annotations are extended with more specific linguistic constructions (e.g., *regular verb forms with infinitive ending in -y*) they cover. This second step uses a rule-based approach with UIMA Ruta (Kluegl et al., 2016) as described by Quixal et al. (2021). The domain model hierarchically associates linguistic constructions with properties, and properties with realizations. Thus, it indirectly links annotated exercises to realizations for which they act as options for adaptive practice. Acquisition of these properties represents discrete steps towards progress completion for a realization. Progress completion is calculated based on interactions with the exercises and pre-defined accuracy thresholds per property. Students' attempts on exercise items are analyzed with respect to correctness, therefore a student's answer is compared to the underlying exercise's target answer. Given that the exercise carries annotations of linguistic property, the interactions with items in the exercise result in either

positive or negative evidence for property acquisition.

Micro-adaptive adjustments while a student works on an exercise, for instance reducing the number of distractors, are not explicitly shown in the dashboard. They are implicitly incorporated in the progress metric as the adaptive algorithm takes the support a student needs into account by weighting the student's attempts respectively.

In existing ITS, the **performance** achieved for a realization is often indicated based on a single exercise, be it the most recent (e.g., Harindranathan and Folkestad, 2019; Britain, 2020) or the best one per realization-inherent difficulty level (e.g., Colling et al., 2022). In our adaptive system, neither of the two makes much sense. Displaying the performance on a single exercise only makes sense if all exercises target similar properties. Since in our implementation, each realization practices various linguistic properties which are distributed over multiple exercises, a single exercise cannot be representative of a student's current performance. Other systems use average performance over all exercises (Keleş et al., 2009). In this approach, performance visualizations of 100% can only be achieved if all answers are correct. However, students might initially provide incorrect answers based on learning gaps or misconceptions which they can overcome in the practice phase. Displaying average performance of all exercises carries the risk of demotivating or even frustrating students as they cannot receive a perfect performance once given a single incorrect answer. In our system, we want to encourage students to also attempt exercises that they cannot master at first try, to benefit from the scaffolding feedback. Pushing students to only work on exercises where they are certain to get everything correct, in order to have a perfectly polished dashboard with 100% in all performance metrics, would be counter-productive for the purpose of learning and practicing in the zone of proximal development, which describes the space of what a learner can acquire when supported (Vygotsky, 1978). Moreover, average performance is not comparable across students and learning target realizations as it does not account for the amount of practice. A metric based on three exercises would put more weight on incorrect solutions than a metric based on 50 exercises. Average values, given in percentages, in general make it less transparent and less intelligible for low literate students to connect the exercise submission

to the performance metrics.

To account for compatibility, transparency and taking learners improvement over time into account, we determine performance by including the most recent ten items instead of focusing on a single submission or an average for an aggregated visualization. As exercises in our system consist of five items, this represents the performance on the last two exercises. Performance is displayed as criterion-referenced performance in a stacked bar chart, giving discrete numbers of items solved correctly at first try, correctly after feedback, and incorrect or not attempted items. This performance display proposed by Colling et al. (2022) shows independent, exam-like as well as scaffolded success and has been evaluated in terms of comprehensibility for seventh graders.

**Mastery criterion** In order to complete the entire learning target, students need to master all its realizations. Mastery is assessed through specific exercises, which we call *diagnostic exercises*. These are manually created by teachers and didacticians and tailored to align well with the practice exercises and the German seventh grade curriculum. Following Colling et al. (2022)'s approach of parallel exercises, there are multiple comparable instances of diagnostic exercises for each realization. This allows students to re-attempt the readiness check after failing a diagnostic exercise. The current diagnostic exercise for a realization is accessible via the `Check readiness` button and assesses the abilities needed to support the functional goal and thus the student's readiness for the target task regarding the particular realization. It takes into account that no support is provided in the communicative task and therefore evaluates only the student's unassisted attempts without providing scaffolding feedback. When a student achieves mastery for a realization by successfully completing its diagnostic exercise, that realization is assigned a trophy symbol. In traditional ITS, readiness to attempt the diagnostic exercise would correspond to having completed all practice exercises of the realization. As there is no predefined sequence of exercises in a macro-adaptive system, in our approach, the adaptive algorithm evaluates, while the student progresses through the adaptive sequence, if the student has practiced all linguistic properties that underlie the realization and if the student's accuracy is at the required proficiency level. Only then can the system reliably predict that the stu-

dent will give a correct solution in the diagnostic exercise. Predicted readiness is made salient by a shiny border around the `Check readiness` button in addition to the full progress pie chart. If a student chooses to work on a diagnostic exercise before the system deems them ready, the system advises them to first practice some more, yet without forcing them to do so. Students are thus guided and scaffolded in the understanding of the provided analytics, in form of progress and performance metrics. This enables students to make sense of their statistical data (van Leeuwen et al., 2022) and as a result identify the next steps towards their learning goal.

**Permanency of mastery** Since traditional ILTS have static exercise sequences, mastery is a permanent attribute. In adaptive systems, however, forgetting needs to be incorporated (Zaidi et al., 2020). In order to consider this in the student dashboard, our trophies "gather dust" once the adaptivity algorithm ascertains that mastery has expired, as demonstrated in Figure 2. This happens if a student hasn't actively – as part of a gap in a gap-filling exercise – or passively – as part of the gaps' co-text – practiced the realization for a set time, which is adjusted based on a student's retention capacities tracked in the learner model. By revising a realization through clicking on the `Check readiness` button – after optionally completing additional practice exercises –, students can prove their maintained proficiency to the system and to themselves. The trophy then regains its shiny appearance.

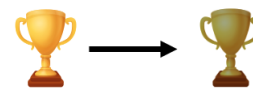


Figure 2: Mastery trophy transition from shiny, i.e., active mastery of respective competence, to dusty, i.e., indicating potential forgetting

**Homework assignment** Teachers who use an ITS as assistant tool for classroom teaching often desire to assign specific exercises to their students which they can then discuss with the entire class (Singh et al., 2011). This is no problem in traditional systems where all exercises are listed and can directly be accessed by the students. In macro-adaptive systems, however, the algorithm dynamically determines the concrete exercise instance

which a student practices. Additionally, in micro-adaptive systems each student receives individually tailored exercises. Consequently, no list of all exercises that is shared amongst all students is available. In order to still facilitate the presented scenario, our system allows teachers to specify mandatory exercises which all students have to complete. By thus giving teachers certain control over the learning content, they can ensure that all students have been exposed to a specific set of hand-selected exercises. The adaptivity algorithm automatically integrates these mandatory exercises into the exercise sequence at the appropriate position according to their associated learning target realizations and the individual student's learning state. However, some students might wish to specifically practice these exercises, either because their slow progress prevents them from reaching the mandatory exercises within the adaptive sequence in a feasible amount of time, or because they do not see the need for additional practice as they might already be proficient in the respective realization. We therefore also explicitly list the mandatory exercises with the option to open them directly. Similarly to the behavior when attempting a diagnostic exercise, if a student chooses to practice a mandatory exercise for which they are not yet proficient enough according to the adaptivity algorithm, the system recommends to first practice more. Students can always decide to ignore these recommendations and proceed to the selected mandatory exercise.

These considerations allow the system to provide common student dashboard features of task-based ITS while also integrating user-adaptivity.

#### 4 Discussion

Since research on simultaneous integration of a task-oriented student dashboard and user-adaptivity in systems applied at secondary school level is very limited, alternative approaches can be considered in some cases and some concepts still lack empirical validation. We therefore discuss potential issues with and alternatives for some of our proposed implementations.

**Diagnostic exercises** In our approach, students can attempt diagnostic exercises by clicking the `Check readiness` button either next to a learning target realization or globally for the entire learning target comprising all its realizations. On the one hand, this global entry point is in line with the adaptive approach requiring low self-regulation. On the

other hand, students using the global button receive all diagnostic exercises in succession so that they do not directly follow the exercises which prepare for them. This decouples the diagnostic exercises for a realization from their scaffolding practice exercises. If the global `Check readiness` button was removed, the question would remain whether all students should proactively attempt the diagnostic exercises themselves – which would potentially result in the same dilemma for students following the adaptive sequence, as they would not be assisted in when to attempt which diagnostic exercise. A solution could be to integrate the diagnostic exercises into the adaptive exercise sequence and saliently flag them for students. Students should then get the choice to attempt the exercise or practice more.

**Transparency** The subject of mandatory exercises leaves an additional question to be addressed. While displaying them globally for the entire learning target avoids the issue of being inaccessible on-demand, it also removes visual assignment to any realization. Since the aim of explicitly listing the realizations is to also foster meta-linguistic knowledge (Godwin-Jones, 2021), neglecting this aspect for mandatory exercises is questionable. Moreover, this would make it harder for students to autonomously reconstruct progress and performance updates from exercise submissions, thus resulting in higher mental load. The lack of transparency in linking exercises to realizations is also an issue for exercises accessed via the global adaptivity buttons `More practice` and `Check readiness` for the entire learning target. This could potentially be addressed by highlighting the associated realization upon opening the exercise.

**Performance visualization** A further discussion point concerns the visualization of student performance. While we have presented an approach to display it as criterion-referenced performance on the most recent items, multiple alternative aggregations and visualizations are envisionable. Representing mastery estimates of concepts (Tong et al., 2022), taking the average performance over multiple exercises, adding up the scores for a defined number of items or only displaying those of the most recent exercise are all valid options (Van Labeke et al., 2007; Harindranathan and Folkestad, 2019). Instead of aggregating multiple exercises, the student dashboard could also visualize all completed exercises for a student individ-

ually. This would also make it possible to mark mandatory exercises. However, if a student practices a lot, the dashboard could quickly become crowded and therefore poorly accessible on small displays (Bull and Kay, 2016). Reducing the exercises to dot representations might increase manageability. Criterion-referenced performance for each exercise could then be displayed on demand after clicking on a dot. In order to increase the discriminability of dot representations, different colors could indicate certain properties of the exercises such as the exercise type, the exercise dimension, or the proficiency level. Making the exercise type salient could for example address varying complexities as for example gap-filling exercises are considered more complex than multiple-choice exercises (Medawela et al., 2018), or varying foci on language dimensions (Grellet, 1981, p. 5) inherent to different exercise types. It would, however, still fail to consider differences in exercise complexity within each exercise type, for instance based on the number of distractors in multiple-choice exercises (Heck et al., 2022). Using the exercise dimensions of receptive, interactive and productive types (Vetter, 2012) instead would reduce the number of categories and thus increase heterogeneity within each category. Since macro-adaptivity aims to gradually increase exercise complexity, the different categories would for both options inadvertently display scores at different stages of the learner's progress, which might not be transparent to students. The alternative approach to associate exercises with the learner's proficiency level at the time of completing them translates continuous proficiency scores of a KT model into concrete categories. Considering the small number of categories, this is also feasible with KT models of moderate accuracy. However, since a student's progress is not always linear (Shirai, 1990) nor are there clear thresholds between the levels, this approach might not give helpful insights either. A compromise between representing all exercises and using a single global aggregation could alternatively collapse exercises with similar colors into a single dot representation with the number of collapsed exercises indicated inside the dot. This would, however, lose the benefits of the non-collapsed representation of highlighting mandatory exercises and providing anchors for criterion-referenced performances per exercise.

**Progress visualization** Although we choose to base our progress measure on linguistic properties, this does not necessarily have to be the case. The categories of exercise type, exercise dimension, and exercise complexity suggested for a performance metric can also be considered for progress. However, categorical progress units, which increase in discrete steps, incorporate ranges of continuous values so that progress does not necessarily increase after each exercise. While KT in principle facilitates continuous and constantly perceivable progress updates, we have already argued that the model's estimates are not accurate enough with insufficient training data.

**Customized learning units** Finally, an adaptive system that supports multiple curricula allows teachers to compile their own learning units. Ideally, teachers can also exclude certain linguistic properties which they do not (yet) wish to practice. Since they may later decide to include these properties, students who have already received a trophy might not fulfill the requirements anymore when also considering the newly included properties. Withdrawing the trophy could be discouraging and the underlying reasoning might not be intuitive to students. A possible solution could use the mechanism of gathering dust so that the trophy would still be visible but inactive. Additionally, the progress pie chart for the realization would have to change accordingly. Making these changes transparent and intelligible for students is not trivial, especially considering that young learners' graphical literacy skills may still be developing (Roberts and Brugar, 2017). It becomes even more of a challenge if multiple learning units practice the same learning target, thus sharing the same pool of exercise candidates. This is especially relevant for teachers who want to incorporate a revision learning target, e.g., having one learning unit where students first learn *simple past*, maybe not including all linguistic properties, but also including *simple past* as a revision when introducing *conditionals type 2* in another learning unit. The question then arises whether performance should be calculated separately within each learning unit – which would hinder the adaptivity algorithm as it would not be able to globally track the students' learning progress – or synchronize progress across the units. Synchronizing progress for realizations where different linguistic properties have been excluded is, however, unfeasible. On the other hand,

disallowing teachers to exclude different properties for different units might also not result in the desired functionalities, especially if teachers intend to practice complementary properties of a realization in different learning units. From a student's perspective, working on a learning target realization in one learning unit but receiving a performance update in another unit as well might lead to misunderstandings, demotivation, or distrust in the technology due to the poor user experience (Franconeri et al., 2021). In the worst case, it could even result in negative learning outcomes. From a teacher's perspective, interpreting and assessing duplicates of identical performance history items in multiple units might be challenging and tedious. Especially the display of mandatory exercises in synchronized learning targets constitutes an open issue.

## 5 Conclusion

We presented the design of a student dashboard for an ITS which integrates curriculum-driven, task-based language teaching and user-adaptivity and has been designed in a co-participatory approach with teachers. We outlined an implementation based on practices and insights from these two instructional approaches that takes into account the opportunities, but also the requirements and restrictions of both. Taking this design as starting point, we critically discussed potential limitations and alternative approaches. Such conceptual and theoretical discussions will guide future work in terms of implementation and evaluation of the dashboard in authentic settings. In a next step, to pilot the design and decide on some open alternatives before fully implementing the dashboard into the system, we plan to evaluate the high-fidelity prototype in a user study with teachers to ascertain efficacy of the design. In this user study we will obtain first quantitative measures on usability and intelligibility. Based on these findings, the refined student dashboard will be implemented using recent front-end development libraries like REACT<sup>1</sup> and build into the modular architecture of the ILTS FeedBook (Parrisius et al., 2022), connecting the dashboard with FeedBook's existing micro-service landscape, including the adaptivity micro-service, the one for NLP processing and the learner model micro-service. The fully implemented dashboard integrated into the ILTS will then be evaluated in a large-scale field study with student participants

<sup>1</sup><https://react.dev/>

using the system in a blended learning setting over an extended period of a school year. The data collected in that study will allow us to identify different learning paths and map them to student characteristics such as high and low self-regulation and navigational preferences such as globally adaptive, realization adaptive or completely self-guided sequencing. Furthermore, the study will yield valuable insights into the practicability and acceptability of the design in real-world usage.

## Acknowledgements

We want to acknowledge the teachers in the AI2Teach project, especially Florian Nuxoll, for their input and feedback.

## References

- Mariam W. Alawar and Samy S. Abu-Naser. 2017. CSS-Tutor: An intelligent tutoring system for CSS and HTML. *International Journal of Academic Research and Development*, 2(1):94–99.
- Katja Andersen. 2019. [Assessing task-orientation potential in primary science textbooks: Toward a new approach](#). *Journal of Research in Science Teaching*, 57:481–509.
- Gabriel Wade Britain. 2020. *Design Analytics Dashboards to Support Students and Instructors*. Ph.D. thesis, Texas A&M University.
- Peter Brusilovsky. 2000. Adaptive Hypermedia: From Intelligent Tutoring Systems to Web-Based Education. In *Intelligent Tutoring Systems*, pages 1–7, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Peter L. Brusilovsky. 1992. A framework for intelligent knowledge sequencing and task sequencing. In *Intelligent Tutoring Systems*, pages 499–506, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Susan Bull, Blandine Ginon, Clelia Boscolo, and Matthew Johnson. 2016. [Introduction of Learning Visualisations and Metacognitive Support in a Persuadable Open Learner Model](#). In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge, LAK '16*, pages 30–39, New York, NY, USA. Association for Computing Machinery.
- Susan Bull and Judy Kay. 2007. Student models that invite the learner in: The smil:() open learner modelling framework. *International Journal of Artificial Intelligence in Education*, 17(2):89–120.
- Susan Bull and Judy Kay. 2010. [Open Learner Models](#). *International Journal of Artificial Intelligence in Education*, 308:301–322.



- Susan Bull and Judy Kay. 2016. [SMILI: a Framework for Interfaces to Learning Data in Open Learner Models, Learning Analytics and Related Fields](#). *International Journal of Artificial Intelligence in Education*, 26(1):293–331.
- Susan Bull and Adam Thomas McEvoy. 2003. An Intelligent Learning Environment with an Open Learner Model for the Desktop PC and Pocket PC. In U. Hoppe, F. Verdejo, and J. Kay, editors, *Artificial Intelligence in Education*, pages 389–391. IOS Press, Amsterdam.
- Jill Burstein, Jane Shore, John Sabatini, Brad Moulder, Steven Holtzman, and Ted Pedersen. 2012. The language musesm system: Linguistically focused instructional authoring. *ETS Research Report Series*, 2012(2):i–36.
- Penghe Chen, Yu Lu, Vincent W. Zheng, and Yang Pian. 2018. [Prerequisite-Driven Deep Knowledge Tracing](#). In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 39–48, Los Alamitos, CA, USA. IEEE Computer Society.
- Jinho D. Choi and Martha Palmer. 2012. [Fast and robust part-of-speech tagging using dynamic model selection](#). In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 363–367, Jeju Island, Korea. Association for Computational Linguistics.
- Leona Colling, Ines Pieronczyk, Stephen Bodnar, Heiko Holz, Cora Parrisius, Katharina Wendebourg, Carolin Blume, Florian Nuxoll, Diana Pili-Moss, and Detmar Meurers. 2022. Practice with a purpose. development of a task-oriented learner dashboard. EUROCALL Conference 2022, virtual.
- Dandi Duan, Antonija Mitrovic, and Neville Churcher. 2010. Evaluating the Effectiveness of Multiple Open Student Models in EER-Tutor. In *Proceedings of the 18th International Conference on Computers in Education*, pages 86–88, Putrajaya, Malaysia. University of Canterbury. Computer Science and Software Engineering.
- Tomáš Effenberger. 2018. *Adaptive system for learning programming*. Ph.D. thesis, Master’s thesis, Masaryk University.
- Rod Ellis. 2016. [Focus on form: A critical review](#). *Language Teaching Research*, 20(3):405–428.
- Mingyu Feng, Jeremy Roschelle, Neil Heffernan, Janet Fairman, and Robert Murphy. 2014. [Implementation of an Intelligent Tutoring System for Online Homework Support in an Efficacy Trial](#). In *"Intelligent Tutoring Systems"*, pages 561–566, Cham. Springer International Publishing.
- David Ferrucci and Adam Lally. 2004. Uima: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3-4):327–348.
- Steven L. Franconeri, Lacey M. Padilla, Priti Shah, Jeffrey M. Zacks, and Jessica Hullman. 2021. The science of visual data communication: What works. *Psychological Science in the public interest*, 22(3):110–161.
- Robert Godwin-Jones. 2021. Big data and language learning: Opportunities and challenges. *Language Learning & Technology*, 25(1):4–19.
- Francoise Grellet. 1981. *Developing Reading Skills: A Practical Guide to Reading Comprehension Exercises*. Cambridge Language Teaching Library. Cambridge University Press, New York.
- Mohamed Hafidi and Tahar Bensebaa. 2014. [Developing Adaptive and Intelligent Tutoring Systems \(AITS\): A General Framework and Its Implementations](#). *International Journal of Information and Communication Technology Education*, 10(4):70–85.
- Priya Harindranathan and James Folkestad. 2019. Learning Analytics to Inform the Learning Design: Supporting Instructors’ Inquiry into Student Learning in Unsupervised Technology-Enhanced Platforms. *Online Learning Journal*, 23(3):34–55.
- Tanja Heck and Detmar Meurers. 2022. [Parametrizable exercise generation from authentic texts: Effectively targeting the language means on the curriculum](#). In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 154–166, Seattle, Washington. Association for Computational Linguistics.
- Tanja Heck, Detmar Meurers, and Florian Nuxoll. 2022. [Automatic exercise generation to support macro-adaptivity in intelligent language tutoring systems](#). In *Intelligent CALL, granular systems and learner data: short papers from EUROCALL 2022*, pages 162–167. Research-publishing.net.
- Alain Hertz. 1992. Finding a feasible course schedule using tabu search. *Discrete Applied Mathematics*, 35(3):255–270.
- Joel A Howell, Lynne D Roberts, Kristen Seaman, and David C Gibson. 2018. Are we on our way to becoming a “helicopter university”? academics’ views on learning analytics. *Technology, Knowledge and Learning*, 23:1–20.
- Aytürk Keleş, Rahim Ocak, Ali Keleş, and Aslan Gülcü. 2009. Zosmat: Web-based intelligent tutoring system for teaching–learning process. *Expert Systems with Applications*, 36(2):1229–1239.
- Peter Kluegl, Martin Toepfer, Philip-Daniel Beck, Georg Fette, and Frank Puppe. 2016. Uima ruta: Rapid development of rule-based information extraction applications. *Natural Language Engineering*, 22(1):1–40.
- Lyn Lim, Maria Bannert, Joep van der Graaf, Shaveen Singh, Yizhou Fan, Surya Surendrannair, Mladen

- Rakovic, Inge Molenaar, Johanna Moore, and Dragan Gašević. 2023. [Effects of real-time analytics-based personalized scaffolds on students' self-regulated learning](#). *Computers in Human Behavior*, 139:107547.
- Qi Liu, Shuanghong Shen, Zhenya Huang, Enhong Chen, and Yonghe Zheng. 2021. [A Survey of Knowledge Tracing](#).
- Tomasz Dominik Loboda, Julio Guerra, Roya Hosseini, and Peter Brusilovsky. 2014. Mastery grids: An open-source social educational progress visualization. In *Proceedings of the 2014 conference on Innovation & technology in computer science education*, pages 357–357.
- Yanjin Long and Vincent Alevan. 2017. [Enhancing learning outcomes through self-regulated learning support with an Open Learner Model](#). *User Modeling and User-adapted Interaction*, 27(1):55–88.
- Roberto Martinez-Maldonado, Abelardo Pardo, Nejin Mirriahi, Kalina Yacef, Judy Kay, and Andrew Clayphan. 2015. Latux: An iterative workflow for designing, validating, and deploying learning analytics visualizations. *Journal of Learning Analytics*, 2(3):9–39.
- R.M. Sumudu Himesha B Medawela, Dugganna Ralalage Dilini Lalanthi Ratnayake, Wijeyapala Abesinghe Mudiyansele Udari Lakshika Abeyasinghe, Ruwan Duminda Jayasinghe, and Kosala Nirmalani Marambe. 2018. [Effectiveness of "fill in the blanks" over multiple choice questions in assessing final year dental undergraduates](#). *Educación Médica*, 19:72–76.
- Guido Minnen, John Carroll, and Darren Pearce. 2001. Applied morphological processing of english. *Natural Language Engineering*, 7(3):207–223.
- Iroda Mirmakhmudova. 2021. [Comparing criterion and norm referenced assessments of language skills in the second language](#). *Asian Journal of Social Sciences and Humanities*, 11:463.
- Robert J. Mislevy, Linda S. Steinberg, and Russell G. Almond. 2002. [Design and analysis in task-based language assessment](#). *Language Testing*, 19(4):477–496.
- Nushi Musa and Hosein Eqbali Mohamad. 2017. Duolingo: A mobile application to assist second language learning. *Teaching English with Technology*, 17(1):89–98.
- Cora Parrisius, Ines Pieronczyk, Carolyn Blume, Katharina Wendebourg, Diana Pili-Moss, Mirjam Assmann, Sabine Beilharz, Stephen Bodnar, Leona Colling, Heiko Holz, et al. 2022. [Using an intelligent tutoring system within a task-based learning approach in english as a foreign language classes to foster motivation and learning outcome \(interact4school\): Pre-registration of the study design](#). PsychArchives.
- Andrea Phillips, John Pane, Rebecca Reumann-Moore, and Oluwatosin Shenbanjo. 2020. [Implementing an adaptive intelligent tutoring system as an instructional supplement](#). *Educational Technology Research and Development*, 68(3):1409–1437.
- Pipatsarun Phobun and Jiracha Vicheanpanya. 2010. Adaptive intelligent tutoring systems for e-learning systems. *Procedia Social and Behavioral Sciences*, 2(2):4064–4069.
- Martí Quixal, Björn Rudzewitz, Elizabeth Bear, and Detmar Meurers. 2021. Automatic annotation of curricular language targets to enrich activity models and support both pedagogy and adaptive systems. In *Proceedings of the 10th Workshop on NLP for Computer Assisted Language Learning*, pages 15–27.
- Kathryn L. Roberts and Kristy A. Brugar. 2017. The view from here: Emergence of graphical literacy. *Reading Psychology*, 38(8):733–777.
- Björn Rudzewitz, Ramon Ziai, Kordula De Kuthy, Verena Möller, Florian Nuxoll, and Detmar Meurers. 2018. [Generating feedback for english foreign language exercises](#). In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 127–136.
- Björn Rudzewitz, Ramon Ziai, Florian Nuxoll, Kordula De Kuthy, and Detmar Meurers. 2019. [Enhancing a Web-based Language Tutoring System with Learning Analytics](#). In *Joint Proceedings of the Workshops of the 12th International Conference on Educational Data Mining co-located with the 12th International Conference on Educational Data Mining, EDM 2019 Workshops, Montréal, Canada, July 2-5, 2019*, volume 2592 of *CEUR Workshop Proceedings*, pages 1–7. CEUR-WS.org.
- Vasile Rus, Dan Stefanescu, Nobal Niraula, and Arthur C. Graesser. 2014. [DeepTutor: Towards Macro- and Micro-Adaptive Conversational Intelligent Tutoring at Scale](#). In *Proceedings of the First ACM Conference on Learning @ Scale Conference, L@S '14*, pages 209–210, New York, NY, USA. Association for Computing Machinery.
- Helmut Schmid. 2005. A programming language for finite state transducers. In *FSMNLP*, volume 4002, pages 308–309. Citeseer.
- Dale H Schunk and Barry J Zimmerman. 2013. Self-regulation and learning. In W. M. Reynolds, G. E. Miller, and I. B. Weiner, editors, *Handbook of psychology: Educational psychology*, pages 45–68. John Wiley & Sons, Inc.
- Beat A. Schwendimann, María Jesús Rodríguez-Triana, Andrii Vozniuk, Luis P. Prieto, Mina Shirvani Boroujeni, Adrian Holzer, Denis Gillet, and Pierre Dillenbourg. 2017. [Perceiving Learning at a Glance: A Systematic Literature Review of Learning Dashboard Research](#). *IEEE Transactions on Learning Technologies*, 10(01):30–41.

- Yasuhiro Shirai. 1990. [U-shaped behavior in L2 acquisition](#). In H. Burmeister and P. L. Rounds, editors, *Handbook of Japanese Psycholinguistics*, volume 2, pages 217–234. De Gruyter, Eugene, OR.
- Ravi Singh, Muhammad Saleem, Prabodha Pradhan, Cristina Heffernan, Neil T. Heffernan, Leena Razzaq, Matthew D. Dailey, Cristine O'Connor, and Courtney Mulcahy. 2011. Feedback during Web-Based Homework: The Role of Hints. In *Artificial Intelligence in Education*, pages 328–336, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Vanja Slavuj, Ana Meštrović, and Božidar Kovačić. 2017. [Adaptivity in educational systems for language learning: a review](#). *Computer Assisted Language Learning*, 30(1-2):64–90.
- Jerry Chih-Yuan Sun, Hsueh-Er Tsai, and Wai Ki Rebecca Cheng. 2023. [Effects of integrating an open learner model with AI-enabled visualization on students' self-regulation strategies usage and behavioral patterns in an online research ethics course](#). *Computers and Education: Artificial Intelligence*, 4:100120.
- Hanshuang Tong, Zhen Wang, Yun Zhou, Shiwei Tong, Wenyan Han, and Qi Liu. 2022. [Introducing Problem Schema with Hierarchical Exercise Graph for Knowledge Tracing](#). In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '22*, pages 405–415, New York, NY, USA. Association for Computing Machinery.
- Carol A. Twigg. 2003. Models for online learning. *Education review*, 38:28–38.
- Kris Van den Branden. 2016. Task-based language teaching. In *The Routledge handbook of English language teaching*, pages 238–251. Routledge, Abingdon, Oxon, UK; New York, NY, USA.
- Nicolas Van Labeke, Paul Brna, and Rafael Morales Gamboa. 2007. Opening up the Interpretation Process in an Open Learner Model. *International Journal of Artificial Intelligence in Education*, 17:305–338.
- Anouschka van Leeuwen, Stephanie D Teasley, Alyssa Wise, Charles Lang, George Siemens, Dragan Gašević, Agathe Merceron, et al. 2022. [Teacher and student facing analytics](#). In *Handbook of Learning Analytics*, pages 130–140. Society for Learning Analytics Research.
- Mieke Vandewaetere and Geraldine Clarebout. 2010. [Perceptions and Illusions about Adaptivity and Their Effects on Learning Outcomes](#). In *Proceedings - 10th IEEE International Conference on Advanced Learning Technologies, ICALT 2010*, pages 480–484, Los Alamitos, CA, USA. IEEE Computer Society.
- Eva Vetter. 2012. [Exploiting receptive multilingualism in institutional language learning: The case of Italian in the Austrian secondary school system](#). *International Journal of Bilingualism*, 16(3):348–365.
- Lev Semenovich Vygotsky. 1978. *Mind in society: Development of higher psychological processes*. Harvard university press.
- Meng Xia, Mingfei Sun, Huan Wei, Qing Chen, Yong Wang, Lei Shi, Huamin Qu, and Xiaojuan Ma. 2019. [Peerlens: Peer-inspired interactive learning path planning in online question pool](#). In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI '19*, page 1–12, New York, NY, USA. Association for Computing Machinery.
- Ahmed Zaidi, Andrew Caines, Russell Moore, Paula Buttery, and Andrew Rice. 2020. Adaptive Forgetting Curves for Spaced Repetition Language Learning. In *Artificial Intelligence in Education*, pages 358–363, Cham. Springer International Publishing.

# GrounDialog: A Dataset for Repair and Grounding in Task-oriented Spoken Dialogues for Language Learning

Xuanming Zhang<sup>1\*</sup>, Rahul Divekar<sup>2</sup>, Rutuja Ubale<sup>2</sup>, and Zhou Yu<sup>1</sup>

<sup>1</sup>Computer Science Department, Columbia University

<sup>2</sup>AI Research Labs, Educational Testing Service

{xz2995, zy2461}@columbia.edu

{rdivekar, rubale}@ets.org

## Abstract

Improving conversational proficiency is a key target for students learning a new language. While acquiring conversational proficiency, students must learn the linguistic mechanisms of Repair and Grounding (R&G) to negotiate meaning and find common ground with their interlocutor so conversational breakdowns can be resolved. Task-oriented Spoken Dialogue Systems (SDS) have long been sought as a tool to hone conversational proficiency. However, the R&G patterns for language learners interacting with a task-oriented spoken dialogue system are not reflected explicitly in any existing datasets. Therefore, to move the needle in Spoken Dialogue Systems for language learning we present GrounDialog: an annotated dataset of spoken conversations where we elicit a rich set of R&G patterns.

## 1 Introduction and Motivation

Many conversations are impromptu back-and-forth interactions that often have no prior preparation or review. As a result, conversational breakdowns (Benner et al., 2021; Li et al., 2020) may occur due to minor misinterpretation, mishearing, mis-speaking, or a general lack of common ground (Traum, 1994). Interlocutors use Repair mechanisms (Albert and de Ruiter, 2018) to detect and resolve communicative problems during conversations; and Grounding mechanisms to establish common ground. For example, we often ask our interlocutors to repeat what they said, explain themselves, request clarifications, etc. Such processes arise proactively or when the initial communication attempt has failed, during which modification and revision to the previous utterances are needed to proceed the conversations naturally.

According to Long (1983), R&G is meaningful in the following perspectives: 1) repair the dis-

\*This work was done while first author was an intern at ETS.

Speaker	Transcriptions
LPS	What um what types presentation is expected?
HPS	I did not understand your last question. Can you be clear?
LPS	<b>I mean what types of presentation would you uh would you expected during the interview?</b>
HPS	You can put up a formal presentation based on your educational background.

Table 1: Example dialogue from GrounDialog. LPS stands for Low-Proficiency Speakers, whereas HPS represents High-Proficiency Speakers.

course when breakdown occurs and 2) avoid conversational breakdowns. Table 1 shows an example dialogue between low-proficiency (LPS) and high-proficiency (HPS) English speakers, where LPS paraphrases themselves to repair the discourse when trouble occurs. Besides, speakers usually try their best to avoid breakdowns in conversations. Based on Long (1983), there are plenty of strategies they can adopt to prevent the breakdowns during communications: 1) relinquish topic control; 2) simplify topic by asking "yes-no" questions; 3) confirm comprehensions of speakers before proceeding, etc.

From the perspective of a language learner, dialogues serve as important media in language acquisition and learning (Eszenyi and van der Wijst, 2006). When language learners chat with high-proficiency speakers, language learners make considerable efforts to ground what they have to say (Eszenyi and van der Wijst, 2006). More specifically, the low-proficiency speakers (LPS) attempt to negotiate the meanings of conversations with high-proficiency speakers (HPS). According to Foster and Ohta (2005) and Cook (2015), interactional processes including negotiation for meaning and various kinds of repair and grounding are among the many ways learners gain access to the second language acquisition. Besides, LPS can also en-

hance their language skills, general communication skills and cultural knowledge during the conversations with HPS (Eszenyi and van der Wijst, 2006).

While R&G is common in nearly all conversations, it is particularly important for language learners as learners are still building up the full understanding of the language. They may also bring R&G influences of their primary language into the language they are learning. It is also possible that low-proficiency speakers (or language learners) employ additional or different R&G mechanisms than high-proficiency speakers of a language. Therefore, there is a lot to know about R&G mechanisms from low-proficiency speakers.

In this paper, we present a dataset that can help linguists and other researchers with several novel linguistic tasks such as identifying R&G patterns. Further, while repair and grounding is an important linguistic mechanism, it is rarely reflected explicitly in the design of spoken dialogue systems that aim to help people learn a new language. Our dataset can fill this gap by allowing researchers to model dialogue state tracking with R&G, generating responses with R&G turns, etc.

We collected this dataset by connecting a high-proficiency speaker and a low-proficiency speaker on a crowd sourcing platform. The high-proficiency speaker played the role of a human resources (HR) assistant in a wizard-of-oz style and was tasked to convey information about an interview. The low-proficiency speaker played the role of an interviewee and was tasked with finding specific information about the same interview through their conversation with the high-proficiency speaker. While R&G may occur as a course of natural conversation, we further induced it by giving the interlocutors some conflicting and incomplete pieces of information. We collected the voice of the low-proficiency speaker and the text responses of the wizard.

To the best of our knowledge, GrounDialog dataset is the first task-oriented dialogue dataset specifically tailored for repair and grounding in spoken conversations between high-proficiency and low-proficiency speakers. Each dialogue in the dataset is transcribed by human experts and contains vocal markers and disfluencies, such as "uh" and "um". It is annotated with R&G types, intents, and slots that are relevant to dialogue state mapping tasks. Hence, GrounDialog can be used to develop a task-oriented conversational agent, equipped with

the R&G ability to detect communicative trouble, and adopt certain strategies to repair the discourse when trouble occurs.

The rest of the paper presents related work, details of the data collection process, the data annotation scheme, analyses of the data, and initial model benchmarks.

## 2 Related Work

As indicated in Dorathy and Mahalakshmi (2011), task-based language teaching (TBLT) puts emphasis on the utilization of tasks as the critical element in the language classroom given that tasks can offer better contexts for active language acquisition and second language promotion. From the perspective of dialogue systems, it is the task-oriented dialogue (ToD) that can help language learners achieve their proficiency goals through task completion. Previous dialogue systems have shown great promise in increasing second language acquisition proficiency. Bibauw et al. (2019) provide an overview of all spoken dialogue systems for language learning. Timpe-Laughlin et al. (2022) have compared learning language via role-play with a spoken dialogue system versus human, and found that spoken dialogue systems are a feasible alternative to human interaction in the role-playing context. Divekar et al. (2021) have found that interaction with spoken dialogue systems in immersive contexts improved students proficiency and decreased their anxiousness while using a foreign language thereby indicating there may be increased willingness to communicate with automated humanoid interlocutors. All this points to evidence that spoken dialogue systems are an effective tool for language acquisition.

Many spoken dialogue systems for the use of language learning have been built using off-the-shelf intent and slot detectors, and dialogue state managers (Bibauw et al., 2019). Divekar et al. (2018) have found some repair and grounding mechanisms in their dialogue system for language learning such as systems being able to respond to learners' questions like "what do you mean" or "what can I say next" in a rule-based system. However, quick scaling up for such systems can only come with datasets.

Several datasets exist to help build task-oriented dialogues such as Schema-Guided-Dialogue (SGD) (Rastogi et al., 2020), MultiWoZ (Budzianowski et al., 2018), Dialogue State Tracking Challenges (DSTC) 1-3 (Williams et al., 2013; Henderson et al.,

2014a,b) and DSTC 4-5 (Kim et al., 2017). Besides, there are other frequently used speech-based ToD data, including Fluent Speech Commands (FSC)<sup>1</sup>, Audio-Snips (Coucke et al., 2018), Carnegie Mellon Communicator Corpus (CMCC)(Bennett and Rudnicky, 2002) and Let’s Go Dataset<sup>2</sup>.

However, existing task-oriented dialogue datasets do not reflect the language learning perspective as there are no constraints in their collection process that one interlocutor must be a low-proficiency speaker. Moreover, most datasets are also a result of a text-based interaction (Wang et al., 2019; Chen et al., 2021; Liang et al., 2021). This also means that the existing datasets will not contain R&G patterns specific for language learners interacting with a task-oriented *spoken* dialogue systems.

Therefore, we present a new dataset, namely GrounDialog, which will be the first dedicated ToD dataset specifically tailored for R&G in HPS-LPS conversations. Besides, the dataset can address the need for R&G in spoken form in specific scenarios that do not exist in the text-based exchange.

### 3 Data Collection Set-up

Our goal was to collect conversations between high-proficiency (HPS) and low-proficiency speakers (LPS). To accomplish this, we use Amazon Mechanical Turk (AMTurk) to recruit and connect pairs of HPS and LPS for our study. To identify whether a participant is HPS or LPS, we provided the participants descriptions of CEFR levels (Council of Europe, 2001) and asked them to self-identify their proficiency level<sup>3</sup>. For the purposes of this study, turkers who identify themselves as *Beginner*, *Elementary*, *Intermediate*, and *Upper Intermediate* i.e., A1-B2 levels were regarded as LPS; whereas those selecting *Advanced* and *Proficient* i.e., C1-C2 are considered as HPS. An assumption of our study is that we draw the line between HPS and LPS arbitrarily at B2 and trust the turker’s self-reported proficiency to be accurate. With this setup, we can end up with nearly equal size of HPS and LPS, which can ease the turker-pairing process for our data collection. A detailed explanation of the data collection process and conversational task for both HPS and LPS is shown below. Subsequently, we

<sup>1</sup><https://fluent.ai/fluent-speech-commands-a-dataset-for-spoken-language-understanding-research/>

<sup>2</sup><https://dialrc.github.io/LetsGoDataset/>

<sup>3</sup>The complete pre-chat survey form is shown in appendix A

will present the general statistics of the collected dialogues and users. The study was approved by the IRB of the institute conducting this research. All participants were adults and provided consent before starting data collection. All collected data released with the paper is anonymized to our best abilities.

#### 3.1 Conversational Task

In order to collect the conversational data that fits our purpose of having a conversation between an automated interlocutor and human, we follow the Wizard-of-Oz set-up (Kelley, 1984). The set-up has also been validated by many previous studies (Wen et al., 2016; Asri et al., 2017; Budzianowski et al., 2018). In general, two turkers (i.e. one HPS and the other LPS) were paired to communicate with each other. We contextualize their task into a pre-interview setting, where an HR hiring manager talks to an interviewee. Specifically, we set LPS to be the interviewee and HPS to be the HR hiring manager. We assign different goals for each role: the interviewee needs to find out the answers to a set of interview-related questions (e.g. interview time, duration, location, etc.), whereas the HR manager is given the information LPS will need and asked to be in charge of scheduling an appointment with the connected interviewee. To induce more repair and grounding turns in the conversation, we provided overlapping but inconsistent information to the interlocutors. For example, the interviewee is instructed that the interview is going to be 30 minutes, whereas the HR manager has 45 minutes in their task specification. We assumed that the difference in information will lead to the interlocutors being confused, asking clarification from each other, and resolving the situation by picking a time (Foster and Ohta, 2005).

#### 3.2 Dialogue Interface

To establish a stable live connection between two turkers, we adapted VisDial AMT Chat (Das et al., 2017) to connect two humans, enable voice input/output, and connect to an off-the-shelf text-to-speech service.

To simulate a Wizard-of-Oz like setting, we enable the LPS to directly record their speech, whereas the HPS input texts into a chat box and their responses are converted into speech using an off-the-shelf Text-To-Speech. The synthesized speech is played on the LPS side. In this way, the LPS could get a feeling of being connected to a

"chatbot", even though the responses are actually written by a human. The instructions for the LPS said that they will be connected to a human or a chatbot. In this way, we left it ambiguous for the LPS to decide for themselves whether they are talking to a chatbot or not. The HPS were told that they would appear as a bot so as to elicit bot-like communication from them. The example dialogue interfaces together with the instructions for both HPS and LPS are shown in Figure 8.

### 3.3 Data Statistics

In total, we collected 42 dialogues, including 1,569 turns, from 55 unique turkers, where there are 29 high-proficiency speakers (HPS) and 26 low-proficiency speakers (LPS). Dialogues collected in our dataset are fairly long, with an average number of 37.4 turns per dialogue. Figure 2 presents a distribution over the sentence lengths for both HPS and LPS. The average sentence lengths are 10.02 and 8.55 for HPS and LPS respectively. We collected a total of 793 spoken utterances from LPS, and 777 textual responses from HPS.

### 3.4 User Statistics

After completing the conversational task, we asked each turker to input their demographic information through a post-chat survey form <sup>4</sup>.

Specifically, for the turkers who did fill in our survey after the chat, there are 35 males and 16 females, with the age spanning from 22 to 63. The majority of the turkers are from India (45%) and the United States (37%). Also, the self-identified English proficiency levels based on CEFR (Council of Europe, 2001) for the collected users are shown in Figure 1. As mentioned before, we take C1-C2 as high-proficiency speaker, and A1-B2 as low-proficiency speaker.

### 3.5 Speech data and transcriptions

There are 793 audio recordings collected from the accepted LPS<sup>5</sup>, of which 586 audio files are transcribed by SpeechPad<sup>6</sup>, a reliable third-party transcription service, and the remaining 207 files are manually transcribed by the lead authors to inspect the quality of the data. The details of the concrete quality inspection process can be found in appendix

<sup>4</sup>Out of 55 unique turkers, four of them did not fill in the post-chat survey.

<sup>5</sup>LPS is accepted based on the speech quality and conversation completeness with HPS.

<sup>6</sup><https://www.speechpad.com/>

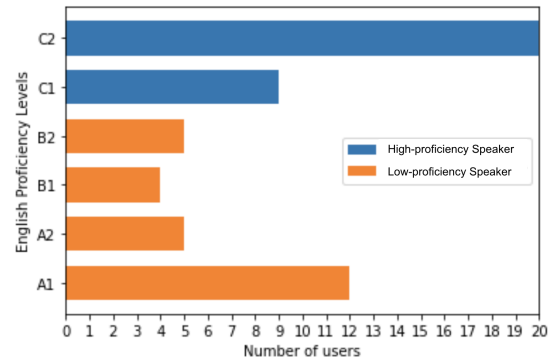


Figure 1: The distribution of CEFR levels in high-proficiency and low-proficiency speakers.

C. The minimum, maximum and mean duration for the audio files collected from LPS are 1.38s, 38.82s and 6.8s, respectively.

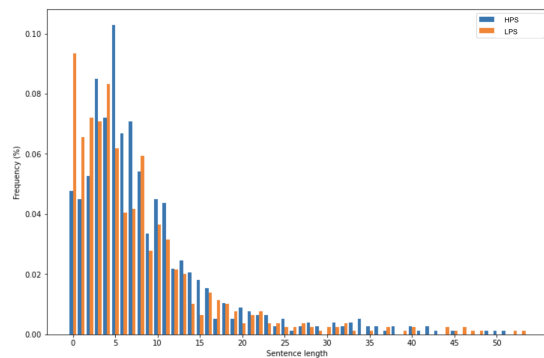


Figure 2: Distribution of number of tokens per turn.

## 4 GroundDialog Corpus

The primary goal of the data collection was to gather free-form conversations with repair and grounding (R&G) patterns, between high-proficiency (HPS) and low-proficiency (LPS) English speakers. For this work, we constrain ourselves to the domain of job interviews, where an HR hiring manager attempts to schedule an upcoming interview with an interviewee candidate and answers any related questions. We leave the conversations in other domains to our future work.

To analyse the R&G patterns in the collected data from MTurk, we inherit R&G types from previous studies (Dobao and Martínez, 2007; Eszenyi and van der Wijst, 2006; Long, 1983; Foster and Ohta, 2005; Schegloff, 1997; Clark, 1996). The complete list of R&G types is shown in table 2. A detailed explanation of R&G annotation scheme is described below. In addition, similar to other task-oriented dialogue datasets (Budzianowski et al.,

2018; Rastogi et al., 2020), we also annotated the intents and slots for our GrounDialog corpus. To ease our annotation process, we adopted Inception (Klie et al., 2018), which is an open-source annotation software platform.

## 4.1 Annotation Scheme

### 4.1.1 Repair and Grounding

R&G can occur over several dialogue turns. It contains the context of the initial communication attempt, questions, and finally a resolution. We tagged these in our dataset as: *Context*, *Question*, *R&G type* and *R&G complete*. The definition for each item type is defined as follows:

- *Context*: the initial utterance as the context of the R&G.
- *Question*: the utterance that triggers the disfluency of the conversation between the two speakers.
- *R&G type*: the R&G type as defined in table 2.
- *Complete*: the utterance that signals the completion of the R&G process.

Note that R&G type is the required item for each R&G annotation, whereas *Context*, *Question* and *Complete* are optional. This is due to the fact that 1) some R&G types can be initiated without the *Context* and *Question* and 2) R&G process maybe not always completed as the conversation moves on.

### 4.1.2 Intent and Slot

Based on the unified dialog acts ontology defined in He et al. (2022), we proposed ontologies for both intent and slot for our GrounDialog corpus. The full ontology is shown in table 3. The more detailed descriptions for each intent and slot are shown in appendix D.

## 4.2 Annotation Statistics and Analysis

### 4.2.1 Repair and Grounding Annotations

The annotations for R&G, Intent and Slot are completed by the lead author. To ensure the quality of the annotations, the lead author and the second author manually inspected each item through comprehensive discussions. The questionable annotation items were corrected if the lead author agreed with the second author.

Figure 3 shows the distribution of different R&G types (a) and R&G related annotations (b) in GrounDialog corpus. There are 269 annotations for R&G types, among which 155 are from HPS and 114 are from LPS. As you can see in figure 3 (a), approximately 30% of the R&G types annotated in HPS utterances are *Proactive Grounding (PG)*. This is due to the fact that the HR manager tends to ask questions that proactively fill in the communication gap and encourage the interviewee candidate to engage in the conversations. For example, in cases when the interviewee candidate forgot to ask questions related to the location of the interview, the HR manager would ask *Do you know how to get to the company?*. On the other hand, as expected, LPS used more *Clarification Request (CR)* in their speech in order to negotiate and confirm critical information for the interview. The example *CR* is shown in table 2.

After including *Context*, *Question* and *R&G complete*, we gathered 604 R&G related annotations, which is nearly 40% of all the dialogue<sup>7</sup>. It can be observed in figure 3 (b) that both HPS and LPS leverage R&G for smoother communication, indicating the potential usefulness of our task set-up in terms of negotiation of meaning in natural HPS-LPS conversations.

### 4.2.2 Intent and Slot Annotation

As for the intent annotations in GrounDialog, there are 1,884 in total, with the number of intents in HPS and LPS being 878 and 1,006, respectively. Figure 4 (left) demonstrates the distribution of intents annotated in the corpus for both HPS and LPS. As you can see, the top two intents are *inform* and *request*, which is similar to larger dialogue datasets like Budzianowski et al. (2018). In our dataset, almost 90% of dialogue utterances have one or two intents indicating the potential of training a language understanding module with our corpus.

Figure 4 (right) presents the distribution of slots annotated in both HPS and LPS responses. There are in total 612 slot annotations, within which 497 slots are annotated from HPS and 115 slots are from LPS. In our GrounDialog corpus, the HPS (i.e. HR managers) tend to give out information in multiple sentences. An example HPS utterance providing concrete location details of the interview is shown below:

<sup>7</sup>Each R&G related annotation is associated with a single utterance. Therefore, the R&G ratio of our dataset is approximately calculated as:  $604 / 1569 \approx 40\%$ .



ID	R&G type	Description	Dialogue Example from GrounDialog
SC	self-correction	When speakers correct own utterances without being prompted to do so by the another person	[Manager]: Are planning to attend the interview? ->Context [Manager]: Are you? -> <b>SC</b>
SP	self-paraphrase	A speaker paraphrases the previous response for another speaker to ensure understanding of the response	[Manager]: You have to make a presentation on Webware as company progressing and about its growth ->Context [Interviewee]: I am sorry I did not get that, could you repeat? ->Question [Manager]: You need to tell us your view about present growth and future growth of company -> <b>SP</b> [Interviewee]: Okay. ->Complete
SR	self-repetition	a speaker repeats the previous utterance given the question from the other speaker due to a communication break	[Manager] The interview will be conducted on Monday next week. ->Context [Interviewee] Sorry I did not get the interview time. Could you repeat that? ->Question [Manager] The interview will be conducted on Monday next week. -> <b>SR</b> [Interviewee] Got it, thanks. ->Complete
SCL	self-clarification	a speaker provides more information as a supplement to their own previous utterances	[Manager] There will be questions about components. ->Context [Interviewee] Yes, ma'am. ->None [Manager] that you find successful -> <b>SCL</b>
QC	question-about -content	a speaker raises question about the contents in the other speaker's response, the contents can include original sentence, phrases, words	[Interviewee] Is there any reimbursement for traveling? ->Context [Manager] reimbursement? -> <b>QC</b>
CU	checking-understanding	the manager asks the interviewee a question to check if they understand what the manager has said	[Manager] The interview will be by Monday next week at 11 am. Will you be able to come? -> <b>CU</b> [Interviewee] Yes, ma'am. ->Complete
CR	clarification-request	One speaker requests for clarification to get some extra information from the other speaker	[Manager] For the interview there will be 5 of us. ->Context [Interviewee] Could you tell me who exactly will be there during the interview? -> <b>CR</b>
TA	tolerate-ambiguity	the manager tolerates the ambiguity in the interviewee's speech and continue the conversation	[Interviewee] Hello. I have some questions about the in-... ->Context [Manager] OK -> <b>TA</b>
RH	recheck-history	the interviewee asks the manager questions that refer back to the dialogue history to recheck the information provided in the conversation	[Interviewee] Just to make sure the interview is on next Monday at 4 pm, right? -> <b>RH</b> [Manager] yes ->Complete
OH	other-help	the manager senses that the interviewee did not finish the previous sentence so the manager provides "acknowledgement" to help the interviewee continue and complete the unfinished utterance	[Interviewee] Hello. Uh ->Context [Manager] Yes please continue -> <b>OH</b> [Interviewee] Uh who will be at the panel? ->Complete
OC	other-correction	the manager finds that the interviewee has made a language mistake and the manager corrects interviewee's mistake	[Interviewee] I want to know is there any green bus meant for traveling? ->Context [Manager] There is no reimbursement. -> <b>OC</b>
PG	proactive-grounding	the speaker proactively grounds the information gap	[Manager] Do you know how to get there? -> <b>PG</b>

Table 2: A full list of R&G types and their descriptions and dialog examples. The R&G annotations for these examples are also shown for each utterance after '->', and the R&G types are highlighted.

<b>Intent type</b>	inform / request / affirm / small_talk thank_you / hi / self_introduction / bye / reqalts / check_connection negate / welcome / not_sure / select / direct / check_availability / propose sorry
<b>Slots</b>	Location / Interview start time / Interview end time / Day / Duration / Interview attendees / Room number / Transportation

Table 3: Full ontology for intent and slot in GrounDialog.

*Ways to commute to our company: from **Penn Station**; exit via **southwest corner** of the station, walk along the **Broadway** for 3 minutes. The company is on the **right side** of the road.*

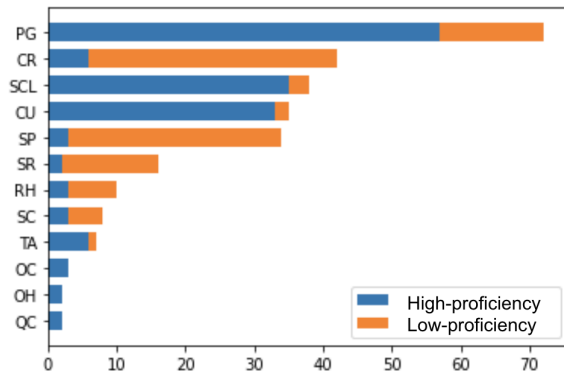
In the example, four values for the *Location* slot are in bold. This is also the reason why nearly 45% of the slots in HPS responses are *Location*. In general, HPS produced much more slots compared

to LPS, which corresponds to the difference in the number of *inform* intent produced in HPS and LPS responses.

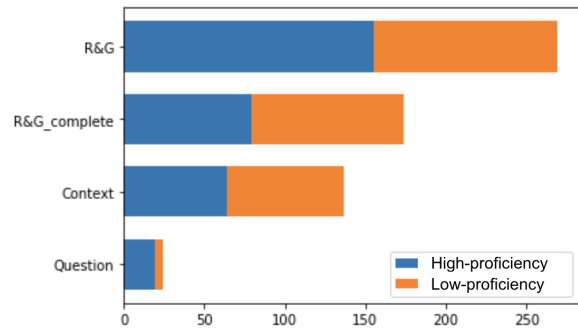
### 4.3 GrounDialog for Language Learning

As the major focus of this work, it is beneficial to take a deeper look at the R&G related annotations in GrounDialog, and discuss the potential utilities of the dataset for language learning.

As we have analyzed in the previous section, nearly 40% of the utterances are related to R&G. Figure 5 also presents the distribution of number of R&G annotations per dialogue. Almost 80% of the dialogues have at least four R&G related annotations, showing the richness of R&G patterns in GrounDialog. In general, GrounDialog encapsulates 12 R&G types in the natural HPS-LPS conversations under our task set-up. According to Figure 3(a), the top three R&G strategies for HPS are *proactive grounding (PG)*, *self-clarification (SCL)* and *check understanding (CU)*, whereas LPS mostly uses *clarification request (CR)*, *self-paraphrase (SP)* and *self-repetition (SR)*. This indicates that GrounDialog explicitly encourages LPS

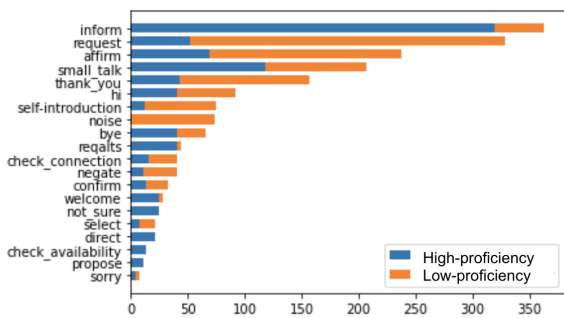


(a) Frequency of R&G types.

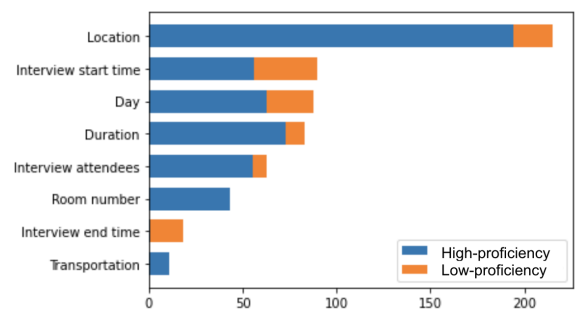


(b) Frequency of R&G related annotations.

Figure 3: Frequency of R&G types (left) and R&G related annotations (right) in GrounDialog.



(a) Frequency of intents.



(b) Frequency of slots.

Figure 4: Frequency of intents (left) and slots (right) in GrounDialog corpus.

to request clarification, rephrase or repeat previous utterances in cases when the initial communication with HPS failed.

Besides, we specifically annotated *R&G complete* to mark the sentences that signals the completion of a R&G process. Based on Figure 3(b), among all 269 R&G annotated in GrounDialog, 174 of them are actually completed, leading to a 65% completion rate. Figure 6 shows the distribution of number of *R&G complete* per dialogue. Nearly 80% of dialogues have at least three *R&G complete*, again suggesting the richness of R&G patterns. Also, given the high frequency of R&G related annotations in figure 3(b), we can imply that HPS tends to initiate the R&G much more often compared to LPS in GrounDialog.

From the language learning perspective, learners need R&G patterns to deepen their understanding of the language. For this purpose, GrounDialog can be used to train a chatbot that can generate responses conditioned on our R&G ontology to initiate R&G process, repair the communication gaps, and ground the meanings of conversations for

the language learners.

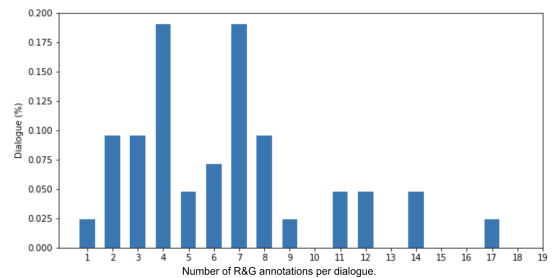


Figure 5: Number of R&G annotations per dialogue.

## 5 GrounDialog as a Benchmark for R&G in Task-oriented Dialogue

GrounDialog is designed as the first dedicated task-oriented dialogue dataset incorporating R&G patterns in HPS-LPS conversations. To show the potential usefulness of the corpus, we break down the dialogue modelling task into two sub-tasks and report a benchmark result for each of them: R&G detection and dialogue state tracking. Specifically, we performed few-shot learning following recent

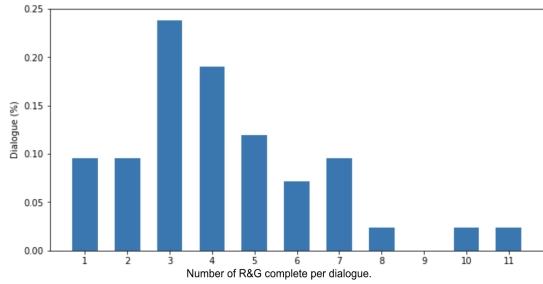


Figure 6: Number of R&G complete per dialogue.

advances in large language models (Brown et al., 2020; Wei et al., 2022), by prompting two most popular large language models, namely ChatGPT and GPT-4<sup>8</sup>, with our carefully engineered prompts for both tasks. The details for each prompt are shown in appendix E.

Model	Slot		Intent	R&G
	Acc	Joint Goal	Acc	Acc
ChatGPT	98.0	88.7	63.1	-
GPT-4	98.2	89.5	65.4	62.1

Table 4: The benchmark results for Dialog State Tracking and R&G detection on GrounDialog.

## 5.1 R&G Detection

We show that by using the R&G annotations in GrounDialog, an R&G detection model can be trained to determine 1) if communication disfluencies occur; and 2) which type of R&G strategy (as defined in table 2) to choose in order to fix the potential disfluencies incurred in conversations.

Similar to previous section, we prompted GPT-4 for this experiment with the specific prompt defined in appendix E. Note that we tested on 40 out of 42 dialogues, excluding the two we used to design the prompt. For the utterances that do not need R&G, we ask the model to predict "None". The overall detection accuracy is shown in table 4 on the rightmost column<sup>9</sup>. As we can see, prompting GPT-4 can achieve over 62% accuracy on the test dialogues, showing the potential of GrounDialog in training neural models in detecting R&G patterns in natural human-human conversations.

<sup>8</sup>We used `gpt-3.5-turbo` for ChatGPT and `gpt-4` (default 8k version) for GPT-4.

<sup>9</sup>We do not report the results for ChatGPT since it failed to follow the prompt instructions.

## 5.2 Dialogue State Tracking

A good conversational system requires robust natural language understanding (NLU) and dialogue state tracking (DST) modules. For our benchmark results, we specifically prompted ChatGPT and GPT-4, both of which are popular ground-breaking large language models (LLMs) these days, with our domain-specific prompts. We follow the evaluation metrics for slot extraction in MultiWoz 1.0 (Budzianowski et al., 2018), where overall slot accuracy and joint goal accuracy are reported. For intent classification, we report the general classification accuracy. Table 4 demonstrates the performance of both models in terms of both sub-tasks. As we have only eight slot types in GrounDialog, both models achieved fairly high scores in slot accuracy and joint goal accuracy, with GPT-4 slightly outperforming ChatGPT. With regard to classifying intents, both models achieved over 60% accuracy, even though we have a larger group of intents to classify. These results demonstrate the potential utility of GrounDialog in building a good task-oriented conversational agent with solid NLU and DST modules.

## 6 Conclusion and Future Work

In this paper, we collected and annotated a new dataset GrounDialog, which is the first dedicated task-oriented dialogue dataset specifically designed for studying repair and grounding in spoken conversations between high-proficiency and low-proficiency speakers. We described the data collection procedure, annotation schemes, and presented a series analysis over the data. In addition, we demonstrated the potential and utility of GrounDialog by performing two tasks: R&G detection and dialogue state tracking. The results showed that GrounDialog can be used to train a conversational agent with the R&G capability. It could be further used to detect communicative gaps, which can be addressed in dialogue design.

In future, we plan to extend GrounDialog to a much larger dataset potentially covering multiple domains other than job interviews. Besides, we will use GrounDialog as a benchmark for a shared task to build task-oriented dialog agent with R&G ability. We will also conduct comprehensive user studies to determine the R&G patterns that are most useful in improving learner’s conversational proficiency during language learning. Further, we plan to present findings from the speech data so

researchers can use speech signals along with text to identify repair and grounding related turns.

## References

- Saul Albert and Jan P de Ruiter. 2018. Repair: the interface between interaction and cognition. *Topics in cognitive science*, 10(2):279–313.
- Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: a corpus for adding memory to goal-oriented dialogue systems. *arXiv preprint arXiv:1704.00057*.
- Dennis Benner, Edona Elshan, Sofia Schöbel, and Andreas Janson. 2021. What do you mean? a review on recovery strategies to overcome conversational breakdowns of conversational agents. In *International Conference on Information Systems (ICIS)*.
- Christina Bennett and Alexander Rudnicky. 2002. The carnegie mellon communicator corpus.
- Serge Bibauw, Thomas François, and Piet Desmet. 2019. Discussing with a computer to practice a foreign language: Research synthesis and conceptual framework of dialogue-based call. *Computer Assisted Language Learning*, 32(8):827–877.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Mićica Gašić. 2018. Multiwoz—a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*.
- Derek Chen, Howard Chen, Yi Yang, Alex Lin, and Zhou Yu. 2021. Action-based conversations dataset: A corpus for building more in-depth task-oriented dialogue systems. *arXiv preprint arXiv:2104.00783*.
- Herbert H Clark. 1996. *Using language*. Cambridge university press.
- Jiyon Cook. 2015. Negotiation for meaning and feedback among language learners. *Journal of Language Teaching and Research*, 6(2):250.
- Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. [Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces](#). *CoRR*, abs/1805.10190.
- Modern Languages Division Council for Cultural Cooperation Council of Europe, Education Committee. 2001. *Common European framework of reference for languages: Learning, teaching, assessment*. Cambridge University Press.
- Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M.F. Moura, Devi Parikh, and Dhruv Batra. 2017. Visual Dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Rahul R Divekar, Jaimie Drozdal, Samuel Chabot, Yalun Zhou, Hui Su, Yue Chen, Houming Zhu, James A Hendler, and Jonas Braasch. 2021. Foreign language acquisition via artificial intelligence and extended reality: design and evaluation. *Computer Assisted Language Learning*, pages 1–29.
- Rahul R Divekar, Jaimie Drozdal, Yalun Zhou, Ziyi Song, David Allen, Robert Rouhani, Rui Zhao, Shuyue Zheng, Lilit Balagyozyan, and Hui Su. 2018. Interaction challenges in ai equipped environments built to teach foreign languages through dialogue and task-completion. In *Proceedings of the 2018 Designing Interactive Systems Conference*, pages 597–609.
- Ana M Fernández Dobao and Ignacio M Palacios Martínez. 2007. Negotiating meaning in interaction between english and spanish speakers via communicative strategies. *Atlantis*, pages 87–105.
- A Anne Dorathy and SN Mahalakshmi. 2011. Second language acquisition through task-based approach—role-play in english language teaching. *English for Specific Purposes World*, 11(33):1–7.
- Réka Eszenyi and Per van der Wijst. 2006. Grounding techniques in computer-mediated classroom tasks. *BELL Belgian Journal of English Language and Literatures*, 4:151–168.
- Pauline Foster and Amy Snyder Ohta. 2005. Negotiation for meaning and peer assistance in second language classrooms. *Applied linguistics*, 26(3):402–430.
- Wanwei He, Yinpei Dai, Yinhe Zheng, Yuchuan Wu, Zheng Cao, Dermot Liu, Peng Jiang, Min Yang, Fei Huang, Luo Si, et al. 2022. Galaxy: A generative pre-trained model for task-oriented dialog with semi-supervised learning and explicit policy injection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10749–10757.
- Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014a. The second dialog state tracking challenge. In *Proceedings of the 15th annual meeting of the special interest group on discourse and dialogue (SIGDIAL)*, pages 263–272.
- Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014b. The third dialog state tracking challenge. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 324–329. IEEE.

- John F Kelley. 1984. An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems (TOIS)*, 2(1):26–41.
- Seokhwan Kim, Luis Fernando D’Haro, Rafael E Banchs, Jason D Williams, and Matthew Henderson. 2017. The fourth dialog state tracking challenge. In *Dialogues with Social Robots*, pages 435–449. Springer.
- Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. [The inception platform: Machine-assisted and knowledge-oriented interactive annotation](#). In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9. Association for Computational Linguistics. Event Title: The 27th International Conference on Computational Linguistics (COLING 2018).
- Toby Jia-Jun Li, Jingya Chen, Haijun Xia, Tom M Mitchell, and Brad A Myers. 2020. Multi-modal repairs of conversational breakdowns in task-oriented dialogs. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, pages 1094–1107.
- Kai-Hui Liang, Patrick Lange, Yoo Jung Oh, Jingwen Zhang, Yoshimi Fukuoka, and Zhou Yu. 2021. Evaluation of in-person counseling strategies to develop physical activity chatbot for women. *arXiv preprint arXiv:2107.10410*.
- Michael H Long. 1983. Native speaker/non-native speaker conversation and the negotiation of comprehensible input. *Applied linguistics*, 4(2):126–141.
- Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.
- Emanuel A Schegloff. 1997. Third turn repair. *AMSTERDAM STUDIES IN THE THEORY AND HISTORY OF LINGUISTIC SCIENCE SERIES 4*, pages 31–40.
- Veronika Timpe-Laughlin, Tetyana Sydorenko, and Judit Dombi. 2022. Human versus machine: investigating l2 learner output in face-to-face versus fully automated role-plays. *Computer Assisted Language Learning*, pages 1–30.
- David R Traum. 1994. A computational theory of grounding in natural language conversation. Technical report, Rochester Univ NY Dept of Computer Science.
- Xuwei Wang, Weiyang Shi, Richard Kim, Yoojung Oh, Sijia Yang, Jingwen Zhang, and Zhou Yu. 2019. Persuasion for good: Towards a personalized persuasive dialogue system for social good. *arXiv preprint arXiv:1906.06725*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.
- Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2016. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*.
- Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. [The dialog state tracking challenge](#). In *Proceedings of the SIGDIAL 2013 Conference*, pages 404–413, Metz, France. Association for Computational Linguistics.

### A Pre-chat English proficiency self-identification survey

See Figure 7 below.

### B Dialogue interface and instructions for High-proficiency and Low-proficiency speakers

See Figure 8 below.

### C Audio data quality inspection

This section details the process to inspect the quality of collected audio data. First of all, due to the fact that some collected audio contains long pauses (usually more than 10 seconds without any valid speech), we listened to each audio that is longer than 15 seconds carefully. Then we used `ffmpeg`<sup>10</sup> to truncate the inspected audio which indeed contains long pause to the extent where the audio is natural and continuous. Next, for each audio data, we applied an internal automatic speech recognition tool to detect if the audio is silent all the time. As a result, we discarded all silent audio, and submit the remaining data to SpeechPad<sup>11</sup> for transcriptions.

### D Descriptions of Intent and Slot

In this section, we explain different types of intent and slots, and show some examples for better understanding. Specifically, we followed the conventions defined in (He et al., 2022). The descriptions for each intent and slot are shown in Table 5 and 6, respectively.

<sup>10</sup><https://ffmpeg.org/>

<sup>11</sup><https://www.speechpad.com/>

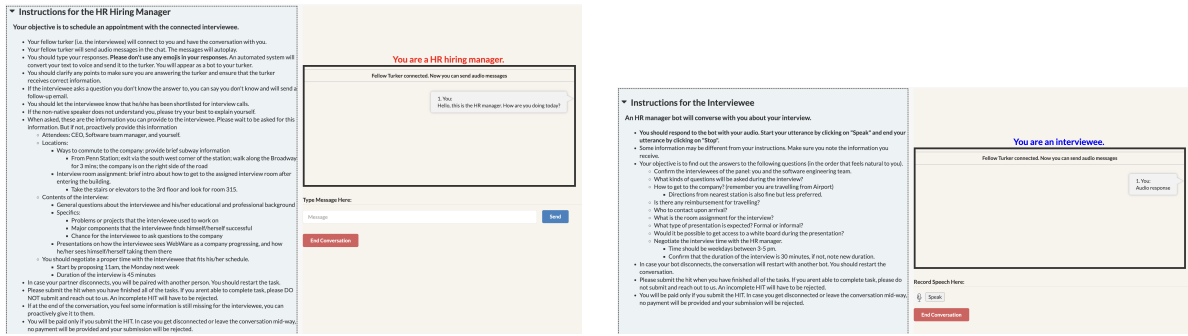
Please fill out the following short survey and submit the information to start the task.

Enter MTurk Worker ID	<input type="text"/>
Please select your highest level of English conversational proficiency (options displayed low to high proficiency)	<input type="radio"/> [Beginner] I can interact in a simple way provided the other person is prepared to repeat or rephrase things at a slower rate of speech and help me formulate what I'm trying to say. I can ask and answer simple questions in areas of immediate need or on very familiar topics. AND I can use simple phrases and sentences to describe where I live and people I know.
	<input type="radio"/> [Elementary] I can communicate in simple and routine tasks requiring a simple and direct exchange of information on familiar topics and activities. I can handle very short social exchanges, even though I can't usually understand enough to keep the conversation going myself. AND I can use a series of phrases and sentences to describe in simple terms my family and other people, living conditions, my educational background and my present or most recent job.
	<input type="radio"/> [Intermediate] I can deal with most situations likely to arise whilst travelling in an area where the language is spoken. I can enter unprepared conversation on topics that are familiar, of personal interest or pertinent to everyday life (e.g. family, hobbies, work, travel and current events). AND I can connect phrases in a simple way in order to describe experiences and events, my dreams, hopes and ambitions. I can briefly give reasons and explanations for opinions and plans. I can narrate a story or relate the plot of a book or film and describe my reactions.
	<input type="radio"/> [Upper Intermediate] I can interact with a degree of fluency and spontaneity that makes regular interaction with native speakers quite possible. I can take an active part in discussion in familiar contexts, accounting for and sustaining my views. AND I can present clear, detailed descriptions on a wide range of subjects related to my field of interest. I can explain a viewpoint on a topical issue giving the advantages and disadvantages of various options.
	<input type="radio"/> [Advanced] I can express myself fluently and spontaneously without much obvious searching for expressions. I can use language flexibly and effectively for social and professional purposes. I can formulate ideas and opinions with precision and relate my contribution skillfully to those of other speakers. AND I can present clear, detailed descriptions of complex subjects integrating sub-themes, developing particular points and rounding off with an appropriate conclusion.
	<input type="radio"/> [Proficient] I can take part effortlessly in any conversation or discussion and have a good familiarity with idiomatic expressions and colloquialisms. I can express myself fluently and convey finer shades of meaning precisely. If I do have a problem I can backtrack and restructure around the difficulty so smoothly that other people are hardly aware of it. AND I can present a clear, smoothly-flowing description or argument in a style appropriate to the context and with an effective logical structure which helps the recipient to notice and remember significant points.
Submit Worker Information	<input type="button" value="Submit"/>

Figure 7: Pre-chat English proficiency self-identification survey.

## E Large Language Models prompts for Dialogue State Tracking and R&G Detection

The prompts we used for experiments in section 5 are shown in Table 7, 8 and 9, respectively. The intent classification and slot extraction task are conducted on a single utterance, whereas R&G detection is conducted on a complete dialogue.



(a) High-proficiency Speaker interface.

(b) Low-proficiency Speaker interface.

Figure 8: Dialogue interface and instructions for connected HPS and LPS.

Intent	Descriptions	Example
hi	greeting responses	"Hello" "How are you"
bye	responses for saying goodbye	"Good bye"
thank_you	responses for appreciation	"Thank you"
welcome	denotes a sentence of official texts to welcome	"Welcome and congratulations! You have been shortlisted for the interview"
small_talk	denotes small chats in daily conversation	"So tell me about yourself" "I am fine"
sorry	apologies responses	"I am sorry."
propose	means suggesting to do/offer/recommend something, in order to make the user consider the performance of a certain action, which the manager believes is in the interviewee's interests.	"How about we meet at 11am on next Monday?"
direct	imperative responses that expresses an order	"You need to arrive early for the interview."
request	asking the user about specific attributes (e.g. duration, location)	"What time of the interview suits your schedule?"
select	asking the user to choose a preferred choice from a set of candidates	"Do you want to do it at 11am or 3pm next Monday?"
reqalts	asking the interviewee for more information	"What else information do you want from me?"
affirm	denotes the affirmative responses	"Yes, there is."
not_sure	means the system is not certain about the interviewee's confirmation	"Sorry, I am not sure about this. I will follow up with an email to confirm later"
negate	denotes the negating responses	"No, it is not"
inform	denotes the normal answers to give the information required by the interviewee	"The duration of the interview is 45 mins"
check_connection	check the connection for the conversation	"Can you hear me?" "There is a lot of background noise"
check_availability	check the availability of the other person	"Are you able to come?" "Are you okay with the timings?"
confirm	confirm to ground information gap	"Shall we set up the interview?"
self-introduction	introduce personal history and past experiences	"I was a software engineer at another company for 2 years ..."

Table 5: Descriptions and examples for each intent type.

Slot	Descriptions	Example
Interview attendees	The attendees of the interview	The <b>CEO, software manager</b> and <b>myself</b> will be in the interview"
Duration	The duration of an event	"The interview duration is <b>45 mins.</b> " "The walking duration is <b>5 mins.</b> "
Room number	The room number of the interview	"Look for room number <b>315.</b> "
Day	Day of the week	"The interview is on next <b>Monday.</b> "
Interview start time	The start time of the interview	"Let's aim for the interview next Monday at <b>3pm</b> "
Interview end time	The end time of the interview	"The interview will end at <b>4pm.</b> "
Location	Any location related information	"Please take the subway to <b>42nd street Time Square</b> " "You should walk along the <b>Broadway.</b> "
Transportation	The transportation mentioned by the speaker	"You will have to travel to our office by <b>train.</b> "

Table 6: Descriptions and examples for each slot type. The slot values are marked in bold.

<p>You need to perform slot extraction tasks. You need to extract "Interview attendees", "Duration", "Room number", "Day", "Interview start time", "Interview end time", "Location", and "Transportation".</p> <p>Or if there is no relevant information, you can output "None".</p> <p>Here are some examples:</p> <p>[Manager] The CEO, software manager and myself will be in the interview -&gt;"Interview attendees": CEO, "Interview attendees": software manager, "Interview attendees": myself [Interviewee] How long is the interview? -&gt;None [Manager] 5 of us will be there in the interview -&gt;None [Manager] There will be 3 of us in the interview -&gt;None [Manager] There will be 3 interviewers during the interview -&gt;None [Manager] The interview will be 45 mins -&gt;"Duration": 45mins [Manager] The walking duration is 3 mins -&gt;"Duration": 3 mins [Manager] Look for room number 315 -&gt;"Room number": 315 [Manager] The interview is on next Monday -&gt;"Day": Monday [Manager] Let's aim for the interview next Monday at 3pm -&gt;"Day": Monday, "Interview start time": 3pm [Interviewee] Can we have the interview between 3 to 5pm instead? -&gt;"Interview start time": 3, "Interview end time": 5pm [Manager] We are scheduling the interview for you on next monday at 4pm. -&gt;"Day": monday, "Interview start time": 4pm [Manager] You will have to travel to our office by train. -&gt;"Transportation": train [Manager] You can take the elevator to the 3rd floor to find the interview room -&gt;"Location": 3rd floor [Manager] Please take the subway to 42nd street Time Square -&gt;"Location": 42nd street Time Square [Manager] way to commute to our company: from Penn station; exit via southwest corner of the station, walk along the broadway for 3 minutes -&gt;"Location": Penn station, "Location": southwest corner of the station, "Location": broadway, "Duration": 3 minutes [Manager] the company is on the right side of the road -&gt;"Location": right side of the road</p> <p>Now, let's predict: [INPUT] -&gt;</p>
---

Table 7: Prompt for slot extraction. The INPUT tag will be replaced with an actual utterance in the dataset during inference.



You need to perform intent classification tasks. Here are the labels and their definitions:

- "hi": greeting responses,
- "bye": responses to say goodbye,
- "thank\_you": responses for appreciation,
- "welcome": welcome and tell the interviewees that they have been shortlisted and selected for interview,
- "small\_talk": small chats in daily conversations,
- "sorry": apologies responses,
- "propose": means suggesting to do/offer/recommend something, in order to make the interviewee consider the performance of a certain action, which the manager believes is in the interviewee's interests,
- "direct": imperative responses that expresses an order,
- "select": manager asks the interviewee to choose a preferred choice from a set of candidates,
- "reqalts": manager asks the interviewee for more information,
- "affirm": denotes the affirmative responses,
- "not\_sure": means the system is not certain about the interviewee's information,
- "negate": denotes the negating responses,
- "inform": denotes the normal answers to give the information required by the interviewee,
- "check\_connection": check the connection for the conversation,
- "check\_availability": check the availability of the other speaker,
- "confirm": confirm to ground information in the chat,
- "self-introduction": interviewee introduces personal history and some past working experiences
- "request-direction": ask about the direction to the company,
- "request-duration": ask about the duration of the interview,
- "request-general-info": ask about the general information,
- "request-interview-attendees": ask about the interview attendees,
- "request-room": ask about the room number of the interview,
- "request-time": ask about the timing of the interview,
- "request-location": ask about the location of the interview,

Or if there is no relevant information, you can output "None".

Here are some examples:

```

Hello ->hi
goodbye ->bye
thank you ->thank_you
[Manager] Welcome and congratulations! ->welcome
[Manager] You have been shortlisted for the interview ->welcome
Tell me about yourself ->small_talk
I am fine ->small_talk
I am sorry ->sorry
okay ->affirm
No, it is not ->negate
We don't have any questions. ->negate
Can you hear me? ->check_connection
There is a lot of background noise ->check_connection
Please tell me more about yourself ->request-general-info
Shall we set up the interview? ->confirm
[Manager] You need to arrive early for the interview ->direct
[Manager] What time of the interview suits your schedule? ->request-time
[Interviewee] How long is the interview? ->request-duration
[Interviewee] How to get to the company? ->request-direction
[Interviewee] How can I find the room of the interview? ->request-room
[Interviewee] Please tell me something ->request-general-info
[Interviewee] Where is the interview? ->request-location
[Interviewee] Who will be there for the interview? ->request-interview-attendees
[Manager] What else information do you want from me? ->reqalts
[Interviewee] I have some background in software development ->self-introduction
[Manager] do you want to do it at 11am or 3pm next Monday? ->select
[Manager] Sorry, I am not sure about this ->not_sure
[Manager] Are you able to come? ->check_availability
[Manager] Are you okay with the timings? ->check_availability
[Manager] Do you know how to get there? ->confirm
[Manager] The duration of the interview is 45 mins ->inform
[Manager] How about next Monday at 11am? ->propose
[Manager] The CEO, Software team manager and I will be meeting with you ->inform
[Manager] You should take subway to Penn Station, exit via the south west corner of the station, walk along the Broadway for 3 mins, and the company is on the right side. ->inform
[Manager] You can take the stairs to 3rd floor and search for room 315. ->inform
[Manager] You need to go to the 3rd floor and find the room. ->inform

```

Now let's predict:  
[INPUT] ->

Table 8: Prompt for intent classification. The INPUT tag will be replaced with an actual utterance in the dataset during inference.

You should extract repair and grounding patterns in the conversations. Here are the labels and their definitions:

- "Context": the initial utterance as the context of the repair and grounding pattern,
- "Question": the utterance that triggers the disfluencies of the conversation between the two speakers,
- "self-paraphrase": a speaker paraphrases the question for another speaker to ensure understanding of the question,
- "checking-understanding": the manager asks the interviewee a question to check if they understand what the manager has said,
- "clarification-request": request for clarification to get some extra information,
- "other-correction": the manager finds that the interviewee has made a language mistake and the manager corrects interviewee's mistake,
- "other-help": the manager senses that the interviewee did not finish the previous sentence so the manager provides "acknowledgement" to help the interviewee continue and complete the unfinished utterance,
- "question-about-content": a speaker raises question about the contents in the other speaker's response, the contents can include original sentence, phrases, words,
- "recheck-history": the interviewee asks the manager questions that refer back to the dialogue history to recheck the information provided in the conversation,
- "self-clarification": a speaker provides more information as a supplement to their own previous utterances,
- "tolerate-ambiguity": the manager tolerates the ambiguity in the interviewee's speech and continue the conversation,
- "proactive-grounding": the speaker proactively grounds the information gap that is not about duration,
- "self-correction": when speakers correct their own utterances without being prompted to do so by another person,
- "self-repetition": a speaker repeats the previous utterance given the question from the other speaker due to communication break,
- "Complete": the utterance that signals the completion of the repair and grounding process and it is normally responding to affirmative questions.

Or if there is no repair and grounding pattern, you can output "None"

Here are two examples for the task: please provide annotation for each utterance below after '->'

Dialogue #1:

[Interviewee] Hallo. ->None  
[Manager] Hi, how are you? ->None  
[Interviewee] I am fine ->None  
[Manager] Good. ->None  
[Manager] Shall we set up the interview? ->proactive\_grounding  
[Interviewee] Yes ->Complete  
[Manager] What do you know about the company's product/services? ->None  
[Manager] What time are you free tomorrow ->self-correction  
[Interviewee] I can only do it from 3 to 5pm. ->None  
[Manager] I see. In that case, do you want to do it at 3pm? ->checking-understanding  
[Interviewee] Yes, I can. ->Complete  
[Manager] Do you know how to get here? ->proactive\_grounding  
[Interviewee] Yes ->Complete  
[Manager] Okay. ->None  
[Manager] there will be questions about components ->None  
[Interviewee] Yes, ma'am. ->None  
[Manager] that you find successful ->self-clarification  
[Manager] The CEO, Software team manager and I will be meeting with you. ->None  
[Interviewee] Okay. ->None  
[Manager] I am sorry, Can you repeat your last response? ->Question  
[Interviewee] I said okay. ->self\_paraphrase  
[Manager] Thank you. ->Complete  
[Interviewee] I need to know how to get to the company ->None  
[Manager] Are you traveling from the airport or train station? ->clarification-request  
[Interviewee] The airport ->Complete  
[Manager] Do you have any questions? ->proactive\_grounding  
[Interviewee] I want to know is there any green bus meant for traveling? ->None  
[Manager] There is no reimbursement. ->other-correction  
[Manager] Then we will see you Monday at 11. ->None  
[Interviewee] Good bye. ->None  
[Manager] bye ->None

Dialogue #2:

[Interviewee] Hello. Uh ->Context  
[Manager] yes please continue ->other-help  
[Manager] Hello I am your hiring manager ->None  
[Interviewee] Hello ->None  
[Manager] I wanted to inform you that you have been shortlisted for an interview ->Context  
[Manager] which will be next week on friday ->self-clarification  
[Manager] What does your wife do? ->Context  
[Interviewee] My wife? ->question-about-content  
[Manager] Yes ->Complete  
[Interviewee] Are we going to have the interview? ->proactive\_grounding  
[Manager] Yes good morning ->Complete  
[Interviewee] How long is the interview? ->None  
[Manager] The duration of the interview will be 45 minutes. ->Context  
[Interviewee] Sorry, I did not catch that. What did you say? ->Question  
[Manager] the duration of the interview will be 45 minutes. ->self-repetition  
[Interviewee] Oh okay. ->Complete  
[Manager] When is your flight? ->Context  
[Interviewee] Sorry, what did you ask? ->Question  
[Manager] At what time will you be leaving for the flight? ->self-paraphrase  
[Interviewee] On Monday 2pm ->Complete  
[Manager] do you have any questions? ->proactive\_grounding  
[Interviewee] How to get to the company? ->None  
[Manager] ways to commute to our company: from Penn station; exit via southwest corner of the station, walk along the broadway for 3 minutes. ->None  
[Manager] the company is on the right side of the road. ->None  
[Interviewee] Okay ->None  
[Interviewee] How can I find the room of the interview? ->None  
[Manager] you will enter the building and look for room 315 on third floor ->None  
[Interviewee] Okay great. ->None  
[Manager] good luck for the interview. Have a great day, bye. ->None  
[Interviewee] Thank you so much. ->None  
[Interviewee] Just to make sure the interview is on next Monday at 4pm, right? ->recheck-history  
[Manager] Yes ->Complete  
[Interviewee] Okay awesome. Thank you ->None  
[Manager] No. Thank you ->None  
[Interviewee] Bye. ->None

Now, please give the prediction for the new conversation. Forget the history and do not generate new dialogue.

[INPUT DIALOGUES]

Table 9: Prompt for R&G detection. The INPUT DIALOGUES tag will be replaced with a complete dialogue during inference.

# SIGHT: A Large Annotated Dataset on Student Insights Gathered from Higher Education Transcripts

Rose E. Wang\*

rewang@cs.stanford.edu

Pawan Wirawarn\*

pawanw@stanford.edu

Noah Goodman

ngoodman@stanford.edu

Dorottya Demszky

ddemszky@stanford.edu

Stanford University

## Abstract

Lectures are a learning experience for both students and teachers. Students learn from teachers about the subject material, while teachers learn from students about how to refine their instruction. However, online student feedback is unstructured and abundant, making it challenging for teachers to learn and improve. We take a step towards tackling this challenge. First, we contribute a dataset for studying this problem: SIGHT is a large dataset of 288 math lecture transcripts and 15,784 comments collected from the Massachusetts Institute of Technology OpenCourseWare (MIT OCW) YouTube channel. Second, we develop a rubric for categorizing feedback types using qualitative analysis. Qualitative analysis methods are powerful in uncovering domain-specific insights, however they are costly to apply to large data sources. To overcome this challenge, we propose a set of best practices for using large language models (LLMs) to cheaply classify the comments at scale. We observe a striking correlation between the model’s and humans’ annotation: Categories with consistent human annotations ( $>0.9$  inter-rater reliability, IRR) also display higher human-model agreement ( $>0.7$ ), while categories with less consistent human annotations ( $0.7$ - $0.8$  IRR) correspondingly demonstrate lower human-model agreement ( $0.3$ - $0.5$ ). These techniques uncover useful student feedback from thousands of comments, costing around \$0.002 per comment. We conclude by discussing exciting future directions on using online student feedback and improving automated annotation techniques for qualitative research.\*

\*Equal contributions.

SIGHT is intended for research purposes only to promote better understanding of effective pedagogy and student feedback. We follow MIT’s Creative Commons License. The dataset should not be used for commercial purposes. We include an elaborate discussion about limitations of our dataset in Section 7 and about the ethical use of the data in the Ethics Statement Section. The code and data are open-sourced here: <https://github.com/rosewang2008/sight>.

## 1 Introduction

Lectures are a learning experience for both students and teachers. Students learn from teachers about the subject material. Teachers also learn from students about how to improve their instruction (for Teaching Project, 2011; Pianta et al., 2008; Evans and Guymon, 1978; Hativa, 1998). However, in the online education setting, student feedback is both abundant and unstructured. This makes it challenging for teachers with online content to synthesize and learn from available feedback.

To take a step towards tackling this challenge, we contribute SIGHT (Student Insights Gathered from Higher Education Transcripts), a large dataset of 288 math lecture transcripts and 15,784 comments collected from the Massachusetts Institute of Technology OpenCourseWare (MIT OCW) YouTube channel. MIT OCW is a popular YouTube channel that offers a collection of lecture content from real MIT courses. Their courses gather up to thousands of student comments (OCW, 2020, 2023; Breslow et al., 2013), in which users express a range of feedback from excitement about the pedagogy to confusion about the course content. The dataset is a rich source of data for studying the relationship between teaching content and student commentary.

Second, we develop a rubric for categorizing different kinds of student feedback in the YouTube comments using a qualitative analysis approach. Qualitative analysis involves iteratively examining the data and accounting for the context (Corbin et al., 1990; Erickson et al., 1985; Bauer and Gaskell, 2000). For example, we examine the student comments for useful feedback categories while accounting for the *online* context of the instruction. Our rubric includes 9 categories of student YouTube comments, spanning from general feedback useful for encouraging instructors (e.g., “Amazing lectures!”) to specific comments on the pedagogy or technical content.

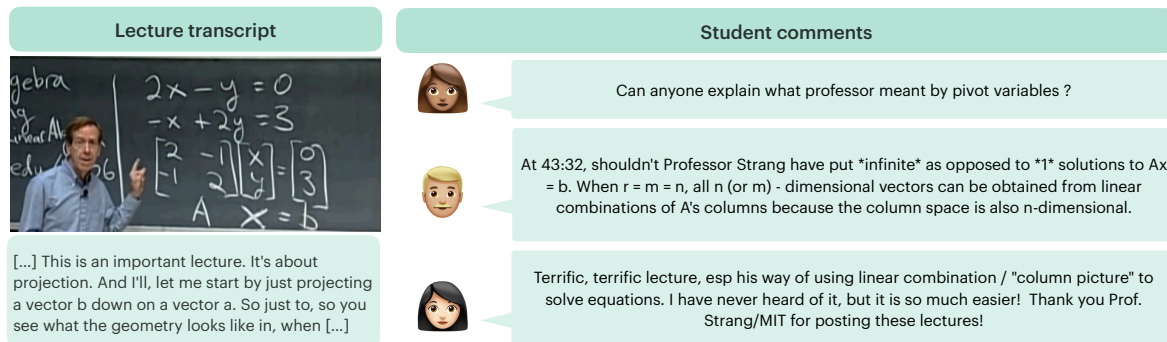


Figure 1: A peek into SIGHT: Every lecture is associated with student comments. SIGHT contains 10 courses, 288 lectures, and 15,784 comments. The comments are labeled using our coding rubric that isolates different types of student feedback.

While qualitative analysis methods are effective in uncovering domain-specific insights, applying these methods to large sources of data is challenging (Erickson et al., 1985; Corbin and Strauss, 1990; Bauer and Gaskell, 2000; O’Connor and Joffe, 2020). Scaling annotation effectively is crucial for sifting through large amounts of unstructured data (e.g., the 15,784 comments in SIGHT) and uncovering relevant student feedback. However, qualitative methodologies often require domain-expertise. This limits the pool of analysts, which makes it expensive to find this expertise or means that only a small sample of the data can be analyzed (Harrison et al., 2019; Lee et al., 2017). Additionally, the qualitative analysis process is time-consuming because it allows for the annotation rubric to be adapted and accommodate new categories. This means that data has to be re-annotated frequently and must be analyzed flexibly. Therefore, our third contribution is proposing a set of best practices for using pretrained large language models (LLMs)—specifically ChatGPT (OpenAI, 2023)—to *cheaply*, *quickly* and *flexibly* annotate data at scale. We explore different prompting approaches (e.g., zero-shot, k-shot, and reasoning).

We analyze the quality of the model annotation and the diversity of user feedback. Categories with consistent human annotations ( $>0.9$  inter-rater reliability, IRR) also display higher human-model agreement ( $>0.7$ ), while categories with less consistent human annotations (0.7-0.8) correspondingly demonstrate lower human-model agreement (0.3-0.5). Albeit imperfect, annotating with ChatGPT allows researchers to explore their entire dataset in a fast, cost-effective way. For example, we are able to sift through 15,784 comments and identify those related to student confusion and lecture pedagogy in a few hours, all under \$0.002 per comment.

These comments can be invaluable for instructors looking to improve their lecture content.

In summary, we make the following contributions in this paper:

1. We create SIGHT, a dataset of 288 lecture transcripts from MIT OpenCourseWare (OCW) mathematics courses and of 15,784 annotated user comments.
2. We develop an annotation rubric of feedback types found in YouTube comments using a qualitative analysis approach.
3. We release a set of best practices for using LLMs with qualitative coding rubrics for scaling annotation.
4. We analyze the quality of the annotation and the diverse types of student feedback uncovered via our automated annotation procedure.

## 2 Related Work

### 2.1 YouTube as an Educational Platform

YouTube is an online platform with a vast collection of educational videos, such as from MIT OCW (OCW, 2023). Due to its popularity and large volume of high-quality education content, YouTube is an important platform for providing educational resources. For example, prior work in education have studied how YouTube provides educational video content to fields like medicine (Curran et al., 2020), support multi-modal learning through video and lecture slides (Lee et al., 2022), or allows for informative discussions in the comment section (Dubovi and Tabak, 2020; Lee et al., 2017). Prior works have not yet studied how YouTube comments can serve as feedback for instructors who host online content.

## 2.2 Student Feedback

Course evaluations by students are the cornerstone for providing feedback to instructors at higher educational institutions (Hammonds et al., 2017; Marsh and Roche, 1997). However, internal course evaluations receive limited student responses, are administered infrequently, and suffer from recency bias (e.g., surveys are typically administered after final examinations) (Cohen, 1981; Greenwald and Gillmore, 1997; Kim and Piech, 2023). Integrating traditional evaluations with informal evaluations from online platforms, like MOOCs and YouTube, can expand the sources of feedback.

Another challenge with course evaluations is that they are unstructured: Student evaluations can encompass a wide range of topics, from technical issues to personal opinions. This is an even more prominent issue for YouTube where videos receive a lot of spam comments. While unsupervised natural language processing (NLP) methods, such as topic modeling, have been applied to survey data to extract themes (Hujala et al., 2020), they struggle to identify specific information. Alternatively, classifiers can be trained for specific domains, e.g., sentiment classifiers for measuring class mood (Hynninen et al., 2019; Baddam et al., 2019; Gottipati et al., 2018; Alhija and Fresko, 2009; Azab et al., 2016), however they are time-consuming to train, especially if the rubrics are modified over the course of analysis. Finally, although qualitative analysis offers powerful insights, it is typically limited to small data samples and is challenging to scale (Asselin et al., 2011; Brook, 2011; Lee et al., 2017).

## 2.3 LLMs for Qualitative Analysis

Recent advances in NLP have resulted in the development of sophisticated pretrained LLMs like ChatGPT (OpenAI, 2023). These models are appealing because they are able to generalize to many domains and follow instructions easily (Brown et al., 2020). We believe these characteristics are particularly appealing for researchers who use qualitative analysis methods and want to explore their dataset fully. Recent works have explored using ChatGPT for annotation on *existing* datasets and benchmarks (Kuzman et al., 2023; Ziems et al., 2023; He et al., 2023; Gilardi et al., 2023). Our work explores applying LLMs to scale annotation on a novel rubric we’ve designed for our research purposes.

Number of lecture series	10
Number of lecture transcripts	288
Number of comments	15,784
Number of labels (Section 4)	9

Table 1: Summary statistics for the SIGHT. We use the labels developed from the coding rubric described in Section 4 to annotate all the comments in the dataset.

## 3 SIGHT

This section details the dataset contents and data collection procedure. Table 1 summarizes the dataset statistics.

### 3.1 Lecture Transcripts

Our work focuses on math lectures from MIT OCW. We use all of the math course playlists listed on MIT OCW as of date, and all of the videos belonging to those playlists. This altogether gives 10 playlists with 288 videos. Each playlist has up to 35 lecture videos. These playlists range from general mathematics courses on calculus and linear algebra to more advanced topics like graph theory and functional analysis. For the full list, please refer to Appendix A.

We use the Google YouTube API to extract the video identification numbers within each playlist, and the YouTube Data API V3 to collect the audio from each video. To transcribe the video audio to text, we use OpenAI’s Whisper large-v2 model (Radford et al., 2022). We manually check the quality of some of the lecture transcripts and find them to be faithful to what is said in the lectures. Our dataset tracks each lecture’s video ID, video title, playlist ID, and transcription model used.

### 3.2 Lecture Comments

We use the Google YouTube API to collect a total of 15,784 user comments from each lecture videos. We do not track the user ID of the comment. If the comment mentions another user with “@”, we anonymize the username by replacing it with “[USERNAME]”. All comments are top-level comments, not replies to comments. This means that if a comment belongs in a thread of another comment, it is not included in our dataset. Each course playlist varies in the number of comments; Appendix A reports the comment statistics. We annotate the comments according to a coding rubric we develop to better understand how users engage with the instruction and lecture content. This rubric is detailed in the next section, Section 4.

Category	Example comment	%	#
general	Best video I have watched so far, I was with him all the way and my concentration never dipped.	28.37%	82
confusion	34:43 why "directional second derivative" would not give us a clue of whether it is a min or max? I thought it is a promising way. hmmm.	20.76%	60
pedagogy	From this lecture, I really understand Positive Definite Matrices and Minima thanks to Dr. Gilbert Strang. The examples really help me to fully comprehend this important subject.	7.27%	21
setup	Oh.. my god.. the board and chalk are phenomenal..!	3.81%	11
personal	sweet, did this like a term and a half ago in highschool. aced the test for it too :D gosh calculus is awesome!	9.00%	26
clarification	@[USERNAME] Actually, if a constant $k=1/1m$ is used, then in the final formula for $V$ you will end up with subtracting $m^1$ from $m^2$ which is apparently not correct.	2.42%	7
gratitude	Thank you very much! Amazing lectures!	13.49%	39
nonenglish	Tłumaczenie na polski wymiata	6.57%	19
na	sounds drunk on 0.5 speed	42.21%	123

Table 2: Example comments for each comment annotation category. The category percentage of the sample dataset is reported in the column `%`. Note, a comment can be labeled with multiple categories so the percentages do not add up to 100%. The number of comments in the sample dataset labelled with that category by at least one of the annotators is reported in the column `#`.

#### Zero-shot prompting for pedagogy category

Consider a YouTube comment from the math MIT OCW video below:

```
Playlist name: {playlistName}
Video name: {videoName}
Comment: {comment}
```

If the statement below is true, please respond "true"; otherwise, please respond "false":  
The comment mentions the teacher's instructional method, which includes but is not limited to the use of examples, applications, worked out problems, proofs, visualizations, elaboration, and analogies.

Figure 2: The zero-shot prompt for the pedagogy category.

## 4 Feedback Rubric

We develop a rubric that catalogs different types of student feedback found in SIGHT. This rubric is used for annotating the comments at scale as well (cf. Section 5). This section details how we developed this rubric.

### 4.1 Rubric Development

When creating the taxonomy, we seek to jointly maximize the following objectives.

- *Provide coverage of feedback expressed in our data.* We uncover categories starting from the data and manually label a subset of the data; this is a part of a qualitative research methodology known as the grounded theory approach (Corbin and Strauss, 1990).

- *Provide coverage of feedback types in the literature.* After we developed a set of categories directly from the data, we consult prior work on course evaluations to incorporate potentially missing themes. Specifically, we used Gravestock and Gregor-Greenleaf (2008); Chen and Hoshower (2003); Zabaleta (2007); Kim and Piech (2023) as additional sources.

- *Be specific about what the feedback is about.* We want to make the feedback categories targeted, enabling instructors to easily understand areas for improvement.

Our process for developing the rubric follows the procedure outlined in O'Connor and Joffe (2020); Seidel et al. (2015); Corbin and Strauss (1990):

Two authors read a subset of randomly selected comments and developed the initial categories collaboratively. The categories were then adapted to be specific and iterated until both authors agreed that the categories sufficiently covered the comments.

## 4.2 Rubric Categories

The final feedback categories in our rubric are detailed below. Examples of each category are shown in Table 2.

**General:** The comment expresses a general/big-picture opinion about the video’s content and/or about the teaching/professional characteristics of the instructor. For example, “Amazing!!!” or “Great teacher.” would be marked as general.

**Confusion:** The comment asks a math-related question, expresses math-related confusion, and/or points out a math-related mistake in the video.

**Pedagogy:** The comment mentions an instructional method. Instructional methods include the use of examples, applications, worked out problems, proofs, visualizations, elaboration, and analogies.

**Teaching setup:** The comment describes or mentions the lecture’s teaching setup. Teaching setup includes the chalk, chalkboard, microphone or audio-related aspects, and camera or camera-related aspects (e.g., angle).

**Personal experience:** The comment mentions the user’s personal experience or context with respect to the lecture. Personal experience or context includes the user’s own math learning or teaching experiences.

**Clarification:** The comment clarifies someone’s math-related misunderstanding or elaborates content from the video, and the comment includes an ‘@’ that is immediately followed by a username.

**Gratitude:** The comment contains the word “thanks” or “thank”.

**Non-English comment:** The comment is not in English.<sup>†</sup>

<sup>†</sup>Because the lectures are conducted in English and the authors feel most comfortable English, we make the distinction between English and non-English comments.

**N/A:** The comment expresses a joke or is a troll comment, and/or the comment says something that is hard to connect to the video content, and/or the comment does not fall into any of the categories above.

## 4.3 Annotation of Sample Dataset

We have two annotators (co-authors) annotate a sample dataset of 280 comments based on the rubric descriptions provided above. The annotators are asked to select all categories that applied. Table 2 reports the category percentage in the sample dataset. Appendix B includes an image of the annotation interface.

## 5 Scaling Annotation

This section details how we scale annotations using our rubric and LLMs. Scaling annotation is crucial for sifting through large amounts of unstructured data (e.g., the 15,784 comments in SIGHT) and uncovering relevant student feedback. By using LLMs, we can cheaply and quickly classify comments, without the need for expensive human annotation.

### 5.1 Model

For scaling annotation, we use GPT-3.5 (gpt-3.5-turbo) through the OpenAI API (OpenAI, 2023). Although alternative models can also be used, such as text-davinci-003 from the original InstructGPT model family (Ouyang et al., 2022), or open-sourced models like Flan-T5 (Chung et al., 2022), GPT-3.5 is cheap, effective, and generally accessible for researchers without GPU support.

### 5.2 Prompting Methods

This section discusses the prompting strategies used for scaling annotation. We also experiment with other approaches, but report the most effective approaches in the main text. We detail our prior attempts and best practices in Appendix D, which we believe to be highly instructive for researchers applying this methodology to other settings.

Each comment is annotated as a binary classification task per category, i.e., does this category apply to this comment? We found that comments oftentimes contained multiple types of feedback. For example, a comment like “His teaching style seems casual and intuitive. I go to a small public college and the course is much more formal

and proof driven. These lectures are a great addition to (as well as a nice break from) formal proofs. Thanks MIT!” includes feedback tied to the pedagogy, personal, and gratitude categories.

**Zero-shot prompting.** Zero-shot prompting directly asks the model to label the category. Following prior work (Child et al., 2019; Ziems et al., 2023), we first provide the context of the comment and the comment itself, then provide instructions on the labelling task. The context of the comment includes a mention to MIT OCW, the playlist name and video name. The instructions include a description of the category, and prompts the model to respond with “true” or “false” for whether the category applies to the comment. Figure 2 shows an example of a zero-shot prompt. The zero-shot setting is the most similar to the human annotation setup.

**K-shot prompting.** K-shot prompting provides examples of the annotation. It first includes the instructions at the top of the prompt, then  $k$  examples that include the context, the comment, and the label. Our work uses 3-shot examples. We did not find any benefits including more than 3 examples. The instructions are moved to the top to avoid repeating the instructions after every example. Due to space constraints, we include our k-shot prompts in Appendix E.

**K-shot prompting with reasoning.** K-shot prompting with reasoning is similar to k-shot prompting, but additionally provides a reasoning for the label. The reasoning comes after the comment, but before the label. Due to space constraints, we include our k-shot reasoning prompts in Appendix E.

### 5.3 Evaluation

We aim to measure the effectiveness of ChatGPT in scaling the annotation process and providing instructors with useful feedback. Our evaluations are centered around the following research questions.

**RQ1:** *How does the zero-shot approach with ChatGPT compare to human annotations across categories?* We investigate this question by measuring the IRR between the human annotations and the zero-shot ChatGPT annotations.

**RQ2:** *How does the incorporation of additional information, such as k-shot examples and reasoning, affect the model’s annotation?* We in-

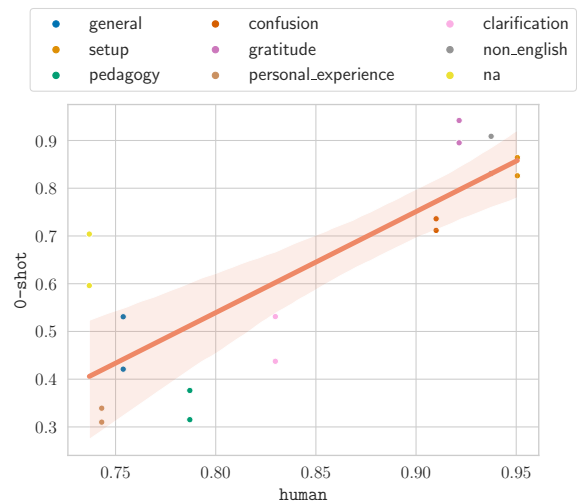


Figure 3: Human inter-annotator agreement (human) vs. human-model inter-annotator agreement ( $\emptyset$ -shot). The agreement scores are color-coded by category.

investigate this question by measuring the IRR between the human annotations and the k-shot and k-shot with reasoning ChatGPT annotations.

**RQ3:** *What are some examples of useful feedback in the scaled annotated dataset?* We investigate this by performing a qualitative analysis (grounded theory approach) on the comments annotated with the confusion category.

## 6 Results

**RQ1.** We compare ChatGPT’s zero-shot annotation to human annotations on the sample dataset described in Section 4.3. We compute Cohen’s kappa to measure IRR within categories. Table 3 reports the human IRR as human and the average human-model IRR on the zero-shot setting as  $\emptyset$ -shot. The human-model agreement never surpasses the human agreement scores. The human-model agreement also varies a lot across categories. For example, the model has fair agreement with the human annotators on pedagogy and personal ( $\sim 0.30$ ) and perfect agreement on gratitude (0.92). To better understand the model’s failure modes, we manually inspect the model’s mislabeled comments. Table 4 shows comments that are representative of common failure modes on pedagogy and personal. The model seems to miss subtle references to either category, such as “explaining their algebra steps” for category pedagogy. The model also tended to mislabel student questions



IRR	gen.	conf.	peda.	set.	pers.	clar.	gra.	noneng.	na
<b>human</b>	0.75	0.91	0.79	0.95	0.74	0.83	0.92	0.94	0.74
<b>0-shot</b>	0.48	0.72	0.35	<b>0.85</b>	0.32	<b>0.48</b>	0.92	<b>0.87</b>	<b>0.65</b>
<b>3-shot</b>	0.50	0.69	0.52	0.75	<b>0.57</b>	0.16	0.85	0.64	0.50
<b>3-shot-R</b>	<b>0.52</b>	<b>0.76</b>	<b>0.57</b>	<b>0.85</b>	0.37	0.32	<b>0.93</b>	0.50	0.47

Table 3: Cohen’s kappa scores for measuring inter-rater reliability (IRR) within humans (human) and within human-model pairs across the rubric categories (abbreviated in the table). We bold the best human-model strategy within each category. The human IRR is used as a reference score. It is in the highlighted row and always reaches substantial to perfect agreement (at least 0.70). The other rows measure the average human-model IRR when the model is prompted 0-shot (0-shot), 3-shot (3-shot), or 3-shot with reasoning (3-shot-R).

#	Category	Comment	H	M
A	pedagogy	This guy is great. I studied engineering at a university less prestigious than MIT, and I remember professors refusing to explain their algebra steps. They were like "you should know this already".	1	0
B	personal	Wish this guy taught me Math 293 and 294 at Cornell. My guy could barely speak English, let alone explain what we were trying to accomplish. I understood that if we wanted eigenvectors perpendicular to x we’d get lift relative to flow...but this guy would have made the math a bit simpler.	1	0
C	pedagogy	41:53 These are questions that should be asked in recitation, not in lecture.	0	1
D	personal	why is iteration in newtons done..i cant understand the logic behind this	0	1

Table 4: Error analysis on pedagogy and personal, the two lowest agreement categories on the zero shot setting (0-shot). The **H** column is the category label that both humans assigned the comment to, and the **M** column is the label that the model assigned the comment to. 1 indicates that the annotator believes the category *does* apply to the comment, whereas 0 is where the category is presumed *not* to apply.

as examples of pedagogy and personal. **Categories that require more interpretation seem to be more difficult for the model to annotate in agreement with humans.**

To further investigate this, we plot human against 0-shot in Figure 3. Strikingly, we observe a correlation between the model’s annotations and the humans’ annotations: Categories exhibiting greater consistency among human annotators (>0.9 IRR) also display higher agreement between humans and the model (>0.7), while categories with less consistent human annotations (0.7-0.8) correspondingly demonstrate lower levels of human-model agreement (0.3-0.5). **Our findings suggest that the model’s annotations reflect the variability observed in human opinions**, providing a complementary perspective to recent works such as He et al. (2023), which report models outperforming humans in annotation tasks. Our results suggest that this superior performance may not always hold, as the model’s annotation accuracy appears to be influenced by the level of human agreement. Appendix C includes additional plots of the category distributions across different annotators.

**RQ2.** The previous section on RQ1’s zero-shot performance indicates that the model poorly annotates categories that involve more qualitative interpretation. Each category has seems to have common failure modes. This section explores potential remedies that provide *more* information to the model: We provide three examples with annotated labels for each category in the prompt. These examples were selected based on the errors the model made in the zero-shot setting. We also test providing the same three examples with human-written reasoning as to why those examples are annotated with such a label.

Table 3 reports the human-model agreement scores on the 3-shot (3-shot) and 3-shot with reasoning setting (3-shot-R). **The effect of the auxiliary information varies across the categories:** Some categories benefit from the examples and reasoning such as pedagogy which does better on 3-shot by +0.17 and on 3-shot-R by +0.22 compared to 0-shot. However, other categories exhibit consistently worse agreement with more information, such as clarification which does worse on 3-shot by -0.32 and on 3-shot-R by -0.16 compared to 0-shot. We also experiment with

Subcategories	Comments labeled as confusion
Conceptual	Can anyone explain what professor meant by pivot variables ?
Conceptual	Can anyone help me understand, why the professor keep saying at 19:01 that we can't solve 4 equation with 3 unknowns?
Potential mistake	i think the explanation of the first question was a little bit wrong it seems. because he wrote the equation to diagonalize the matrix P even though it does not have 3 independent eigen vectors.
Potential mistake	Anyone understand the equation at 32:15? I think $x_{free}$ should be above $x_{pivot}$ ?
Resources	What is good homework to test if we clearly understand this lecture? Is there such corresponding homework?
Resources	Does anyone know which lecture he derive the general equation for a determinant? Would be a massive help thanks!

Table 5: Example comments in the confusion category.

selectively picking the examples and tuning the reasoning, but those attempts did not result in better agreement across the categories. We flag this as an important area to address by future research for performing annotation work with LLMs.

**RQ3.** This section explores the annotated dataset on confusion, a category that has high human-model IRR and would be useful for providing instructors feedback. Of the total 15,784 comments, about 16% are annotated with confusion by the model. The model allows for *easy* identification of areas where students are struggling to understand the material. This information can be invaluable for teachers looking to refine their instruction (e.g., minimize confusion in their teaching material) and improve the learning experience for their students. Table 5 illustrates the diversity in subcategories within confusion. We used the qualitative research approach of grounded theory to discover these categories. There is a range of comments which ask a conceptual questions (e.g., “Can anyone explain what professor meant by pivot variables ?”) or express confusion due to a potential mistake in the lecture (e.g., “i think the explanation of the first question was a little bit wrong it seems.”) Instructors may use these identified comments to appropriately adapt their lecture content in future course iterations.

## 7 Limitations

While our work provides a useful starting point for understanding student feedback, there are limitations to our work. Addressing these limitations will be an important area for future research.

**Comments may not reflect real student feedback.** The comments in our dataset are from users who have chosen to post publicly on YouTube. Addi-

tionally, the comments may include features specific to this online education setting. Thus, the comments may reflect real student comments from these courses.

**There is a selection bias in lecture sources.** SIGHT includes lectures that may be drawn from the most successful offerings of that course. The instructional quality may not be representative of typical instruction. Thus, inferences drawn about the instruction should be interpreted with caution, as they might not generalize to other lecture settings.

**We analyze only English comments.** We analyze only English comments because the lecture content is given in English and the authors are most comfortable with English. As a result, our rubric may not capture the types of feedback from non-English students watching lectures taught in English. In the future, the rubric and analysis should be adapted to account for the multilingual feedback setting.

**We annotate a small subsample of the data** To assess the validity of the automatic labels, we conduct a diagnostic study on a small, randomly selected subset of the dataset, comprising approximately 2% of the comments. Our work aims to establish a preliminary evaluation of the human-model agreement and model annotations, and further validation of the automatic labels is necessary. Future work can focus on acquiring such gold-standard annotations to enhance the quality and reliability of the automatic labels.

## 8 Future Work

This work contributes SIGHT (a dataset of lecture transcripts and student comments), a rubric for annotating student comments, and an analysis on the

annotation quality of LLMs and annotated comments.

**Synthesizing student feedback effectively for instructors.** Given the large volume of feedback that instructors receive, it is important to develop methods for summarizing student feedback (Hu et al., 2022). Equally important is how to present the feedback to teachers such that teachers receive it well and can easily incorporate it into their instruction (Yao and Grady, 2005; Lindahl and Unger, 2010).

**Revising the lecture content with respect to student feedback.** SIGHT contains per-lecture user comments. This can serve as *language feedback* for revising the lecture content conditioned on student feedback (Scheurer et al., 2022).

**Expanding the human annotations** Our work relies on two annotators (co-authors) familiar with rubric categories who annotated 280 comments from the total of 15,784 comments. Future work can investigate expanding the human annotations using our rubric, which may be useful for finetuning or evaluations. Additionally, the rubric categories focus on the themes that emerge from the comments. These can act as an initial filter on relevant versus irrelevant comments for instruction feedback. Future work can consider incorporating categories that play a more specific role for their use case, such as capturing the student’s experience in the course (Welch and Mihalcea, 2016; Ganesh et al., 2022).

**Improving the model’s annotation for qualitative analysis methods** Our work shows that the model does not annotate categories well that require more interpretation, even with auxiliary information. Future work can explore alternative best practices needed in prompting for these types of categories.

## 9 Conclusion

Our work contributes SIGHT, a large-scale dataset of lecture transcripts and student comments. We propose a rubric and different prompting methods for performing automated annotations on SIGHT. While we find that there is still room for improvement on reliably automating the annotation process, the dataset and rubric provide a foundation for future research to address the challenges of discovering useful feedback from students at scale.

For qualitative researchers, SIGHT offers a unique opportunity to investigate and gain insights into the feedback provided by students in online learning environments. The comments cover students’ perspectives and opinions related to math lectures. Educators can also leverage SIGHT as a valuable resource to learn from student comments and refine their teaching materials. By analyzing the feedback provided by students, educators can identify strengths and weaknesses in their instruction, discover areas that students find challenging or confusing, and gather valuable insights to enhance their teaching methodologies. We hope the dataset and methods building off of this dataset can aid educators in making data-informed decisions to optimize their instructional practices, thereby promoting a more effective learning environment.

## Ethics Statement

The purpose of this work is to promote better understanding of effective pedagogy and student feedback through the use of NLP techniques. The SIGHT dataset is intended for research purposes only, and is licensed under MIT’s Creative Commons License. The dataset should not be used for commercial purposes, and we ask that users of our dataset respect this restriction. As stewards of this data, we are committed to protecting the privacy and confidentiality of the individuals who contributed comments to the dataset. It is important to note that inferences drawn from the dataset may not necessarily reflect the experiences or opinions of real students, and should be interpreted with caution. The intended use case for this dataset is to further education research and improve teaching and learning outcomes. Unacceptable use cases include any attempts to identify users or use the data for commercial gain. We additionally recommend that researchers who do use our dataset take steps to mitigate any risks or harms to individuals that may arise.

## Acknowledgements

REW is supported by the National Science Foundation Graduate Research Fellowship. We thank Professor Shannon Seidel for feedback during the initial stages of the project. We thank Yunsung Kim, Allen Nie, Haley Lepp, and Gabriel Poesia for their feedback on the manuscript. We also thank Betsy Rajala, Quinn Waeiss and Professor Michael Bernstein from the Ethics and Society Review Board for

guidance on the ethical and safety risks with the data used in our work.

## References

- Fadia Nasser-Abu Alhija and Barbara Fresko. 2009. Student evaluation of instruction: what can be learned from students' written comments? *Studies in Educational evaluation*, 35(1):37–44.
- Marlene Asselin, Teresa Dobson, Eric M Meyers, Cristina Teixeira, and Linda Ham. 2011. Learning from youtube: an analysis of information literacy in user discourse. In *Proceedings of the 2011 iConference*, pages 640–642.
- Mahmoud Azab, Rada Mihalcea, and Jacob Abernethy. 2016. Analysing ratemyprofessors evaluations across institutions, disciplines, and cultures: The tell-tale signs of a good professor. In *Social Informatics: 8th International Conference, SocInfo 2016, Bellevue, WA, USA, November 11-14, 2016, Proceedings, Part I 8*, pages 438–453. Springer.
- Swathi Baddam, Prasad Bingi, and Syed Shuva. 2019. Student evaluation of teaching in business education: Discovering student sentiments using text mining techniques. *e-Journal of Business Education and Scholarship of Teaching*, 13(3):1–13.
- Martin W Bauer and George Gaskell. 2000. *Qualitative researching with text, image and sound: A practical handbook for social research*. Sage.
- Lori Breslow, David E Pritchard, Jennifer DeBoer, Glenda S Stump, Andrew D Ho, and Daniel T Seaton. 2013. Studying learning in the worldwide classroom research into edx's first mooc. *Research & Practice in Assessment*, 8:13–25.
- Jennifer Brook. 2011. The affordances of youtube for language learning and teaching. *Hawaii Pacific University TESOL Working Paper Series*, 9(1):2.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Yining Chen and Leon B Hoshower. 2003. Student evaluation of teaching effectiveness: An assessment of student perception and motivation. *Assessment & evaluation in higher education*, 28(1):71–88.
- Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. [Generating long sequences with sparse transformers](#).
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. [Scaling instruction-finetuned language models](#).
- Peter A Cohen. 1981. Student ratings of instruction and student achievement: A meta-analysis of multisec-tion validity studies. *Review of educational research*, 51(3):281–309.
- Juliet Corbin et al. 1990. Basics of qualitative research grounded theory procedures and techniques.
- Juliet M Corbin and Anselm Strauss. 1990. Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative sociology*, 13(1):3–21.
- Vernon Curran, Karla Simmons, Lauren Matthews, Lisa Fleet, Diana L Gustafson, Nicholas A Fairbridge, and Xiaolin Xu. 2020. Youtube as an educational resource in medical education: a scoping review. *Medical Science Educator*, 30:1775–1782.
- Ilana Dubovi and Iris Tabak. 2020. An empirical analysis of knowledge co-construction in youtube comments. *Computers & Education*, 156:103939.
- Frederick Erickson et al. 1985. *Qualitative methods in research on teaching*. Institute for Research on Teaching.
- Warren E Evans and Ronald E Guymon. 1978. Clarity of explanation: A powerful indicator of teacher effectiveness.
- Learning Mathematics for Teaching Project. 2011. Measuring the mathematical quality of instruction. *Journal of Mathematics Teacher Education*, 14(1):25–47.
- Ananya Ganesh, Hugh Scribner, Jasdeep Singh, Katherine Goodman, Jean Hertzberg, and Katharina Kann. 2022. [Response construct tagging: NLP-aided assessment for engineering education](#). In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 250–261, Seattle, Washington. Association for Computational Linguistics.
- Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. [Chatgpt outperforms crowd-workers for text-annotation tasks](#).
- Swapna Gottipati, Venky Shankararaman, and Jeff Rongsheng Lin. 2018. Text analytics approach to extract course improvement suggestions from students' feedback. *Research and Practice in Technology Enhanced Learning*, 13:1–19.
- Pamela Gravestock and Emily Gregor-Greenleaf. 2008. *Student course evaluations: Research, models and trends*. Higher Education Quality Council of Ontario Toronto.

- Anthony G Greenwald and Gerald M Gillmore. 1997. Grading leniency is a removable contaminant of student ratings. *American psychologist*, 52(11):1209.
- Frank Hammonds, Gina J Mariano, Gracie Ammons, and Sheridan Chambers. 2017. Student evaluations of teaching: improving teaching quality in higher education. *Perspectives: Policy and Practice in Higher Education*, 21(1):26–33.
- Colin D Harrison, Tiffy A Nguyen, Shannon B Seidel, Alycia M Escobedo, Courtney Hartman, Katie Lam, Kristen S Liang, Miranda Martens, Gigi N Acker, Susan F Akana, et al. 2019. Investigating instructor talk in novel contexts: Widespread use, unexpected categories, and an emergent sampling strategy. *CBE—Life Sciences Education*, 18(3):ar47.
- Nira Hativa. 1998. Lack of clarity in university teaching: A case study. *Higher Education*, pages 353–381.
- Xingwei He, Zhenghao Lin, Yeyun Gong, A-Long Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, and Weizhu Chen. 2023. [Annollm: Making large language models to be better crowdsourced annotators.](#)
- Yinuo Hu, Shiyue Zhang, Viji Sathy, AT Panter, and Mohit Bansal. 2022. Setsum: Summarization and visualization of student evaluations of teaching. *arXiv preprint arXiv:2207.03640*.
- Maija Hujala, Antti Knutas, Timo Hynninen, and Heli Arminen. 2020. Improving the quality of teaching by utilising written student feedback: A streamlined process. *Computers & education*, 157:103965.
- Timo Hynninen, Antti Knutas, Maija Hujala, and Heli Arminen. 2019. Distinguishing the themes emerging from masses of open student feedback. In *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 557–561. IEEE.
- Yunsung Kim and Chris Piech. 2023. High-resolution course feedback: Timely feedback for course instructors.
- Taja Kuzman, Igor Mozetic, and Nikola Ljubešić. 2023. Chatgpt: Beginning of an end of manual linguistic data annotation? use case of automatic genre identification. *arXiv e-prints*, pages arXiv–2303.
- Chei Sian Lee, Hamzah Osop, Dion Hoe-Lian Goh, and Gani Kelni. 2017. Making sense of comments on youtube educational videos: A self-directed learning perspective. *Online Information Review*, 41(5):611–625.
- Dong Won Lee, Chaitanya Ahuja, Paul Pu Liang, Sanika Natu, and Louis-Philippe Morency. 2022. Multi-modal lecture presentations dataset: Understanding multimodality in educational slides. *arXiv preprint arXiv:2208.08080*.
- Mary W Lindahl and Michael L Unger. 2010. Cruelty in student teaching evaluations. *College Teaching*, 58(3):71–76.
- Herbert W Marsh and Lawrence A Roche. 1997. Making students’ evaluations of teaching effectiveness effective: The critical issues of validity, bias, and utility. *American psychologist*, 52(11):1187.
- MIT OCW. 2020. 2020 ocw impact report. [https://ocw.mit.edu/ocw-www/2020-19\\_ocw\\_impact\\_report.pdf](https://ocw.mit.edu/ocw-www/2020-19_ocw_impact_report.pdf).
- MIT OCW. 2023. Massachusetts institute of technology: Mit opencourseware. <https://ocw.mit.edu/>.
- OpenAI. 2023. Introducing chatgpt and whisper apis. <https://openai.com/blog/introducing-chatgpt-and-whisper-apis>.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback.](#)
- Clíodhna O’Connor and Helene Joffe. 2020. Intercoder reliability in qualitative research: debates and practical guidelines. *International journal of qualitative methods*, 19:1609406919899220.
- Robert C Pianta, Karen M La Paro, and Bridget K Hamre. 2008. *Classroom Assessment Scoring System™: Manual K-3*. Paul H Brookes Publishing.
- Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. [Robust speech recognition via large-scale weak supervision.](#)
- Jérémy Scheurer, Jon Ander Campos, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. 2022. [Training language models with language feedback.](#)
- Shannon B Seidel, Amanda L Reggi, Jeffrey N Schinske, Laura W Burrus, and Kimberly D Tanner. 2015. Beyond the biology: A systematic investigation of non-content instructor talk in an introductory biology course. *CBE—Life Sciences Education*, 14(4):ar43.
- Charles Welch and Rada Mihalcea. 2016. [Targeted sentiment to understand student comments.](#) In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2471–2481, Osaka, Japan. The COLING 2016 Organizing Committee.
- Yuankun Yao and Marilyn L Grady. 2005. How do faculty make formative use of student evaluation feedback?: A multiple case study. *Journal of Personnel Evaluation in Education*, 18:107–126.

Francisco Zabaleta. 2007. The use and misuse of student evaluations of teaching. *Teaching in higher education*, 12(1):55–76.

Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. 2023. Can large language models transform computational social science? *arXiv submission 4840038*.

## A SIGHT details

The playlists used in SIGHT are:

- 6.041 Probabilistic Systems Analysis and Applied Probability
- 18.01 Single Variable Calculus
- 18.02 Multivariable Calculus
- 18.404J Theory of Computation
- 18.06 Linear Algebra
- 18.065 Matrix Methods in Data Analysis, Signal Processing, and Machine Learning
- 18.100A Real Analysis
- 18.102 Introduction to Functional Analysis
- 18.217 Graph Theory and Additive Combinatorics
- 18.650 Statistics for Applications

Table 6 shows each playlist’s number of comments. These comments are collected through Google’s YouTube API as detailed in Section 3.

## B Human annotation interface

Figure 6 shows an example of what the annotation interface looks like. Each comment had to be labeled with at least one category in order to proceed to the next comment. The human annotators annotated the same comments. A total of 290 comments are manually annotated and are used to perform the annotation analysis in Section 6.

## C Distribution of annotations

Figure 4 shows the distribution of comment categories from the two human annotators and model.

## D Scaling annotation

This section documents prior attempts at scaling annotation with LLMs. We believe this is highly instructive for researchers applying LLMs to other domains to facilitate their qualitative analysis.

## D.1 Multi-class classification with entire rubric

**Setup** We first attempted to perform multi-class classification over all 9 categories on each comment, i.e., which one category best applies to this comment? Note this is different from the multi-label classification scheme that our work performs in the main text. This method similarly provides the context of the comment and the comment itself, then provide instructions on the labelling task. The context of the comment includes a mention to MIT OCW, the playlist name, and the video name. The instructions include the entire annotation rubric: a list of the all category names and descriptions. It ends by instructing the model to respond with the category that best applies to the comment. An example of such a multi-class classification prompt is shown in Figure 7.

**Results** First, the human-model agreement scores were generally moderate ( $\sim 0.50$ ). We found that the **model did not follow constraints we had set for some categories**. One constraint is on gratitude: label comments as gratitude if and only if they contain “thanks” or “thank”. We found that the model would still label comments that alluded to being grateful but did not follow this constraint as gratitude. Attempts at tuning the prompt did not result in higher IRR.

Additionally, **a comment may belong to multiple categories in our rubric, making it challenging to make the model to just the one best category for the comment**. The human annotators did have rules for resolving category conflicts, i.e., when a comment belongs to more than one category, which of the categories to assign. These rules were also included in the prompt, however this did not improve the model’s annotations much.

## D.2 Staged divide-and-conquer multi-class classification

**Setup** To more directly help LLMs choose the one label that best applies to a comment, we give the LLM a series of simpler classification subtasks, i.e. a staged classification scheme. In the first stage, we ask the model to annotate categories that are more easily resolvable, e.g., gratitude, which only looks for the words “thanks” or “thank” in comments. We provide the model the option to also annotate with “none” if none of the of categories in the stage apply. These “none” comments then transition to the next stage of labelling. The

Playlist name	Number of comments
6.041 Probabilistic Systems Analysis and Applied Probability	1,031
18.01 Single Variable Calculus	3,293
18.02 Multivariable Calculus	2,642
18.404J Theory of Computation	202
18.06 Linear Algebra	6,021
18.065 Matrix Methods in Data Analysis, Signal Processing, and Machine Learning	1,448
18.100A Real Analysis	244
18.102 Introduction to Functional Analysis	129
18.217 Graph Theory and Additive Combinatorics	78
18.650 Statistics for Applications	696

Table 6: Number of comments from each playlist in our dataset.

following stages ask the LLM to classify between a set of categories that are often mistaken for each other, and continue to pass comments that have not gotten a label to the following stage(s). One example of categories that the LLM frequently mistakes for each other are `general` and `pedagogy`: Although both categories can include comments about the teacher, `pedagogy` should only include comments that explicitly talk about the teacher’s instructional method, while `general` should include comments that expresses more general opinions about the teacher. Therefore, we tried grouping `general` and `pedagogy` together in one stage in hopes of helping the model more clearly see the difference between the two categories.

**Results** We found that compared to the previous classification scheme (Section D.1), **asking the LLM to classify between a smaller set of categories that are often mistaken for each other does reduce the number of errors that the LLM makes between those categories.** For example, from frequently mistaking `pedagogy` as `general` comments when prompted with the entire rubric (i.e., all other categories), isolating the classification between just `general` and `pedagogy` to one stage helped the LLM more accurately decide between labelling a comment as `general`, `pedagogy`, or none of the two. However, the human-model agreement scores were still at most moderate ( $\sim 0.60$ ).

Our attempts to perform multi-class classification (with the entire rubric and with the staged divide-and-conquer method) led us to hypothesize that in our setting, the task of choosing only one label that best applies to a comment is too difficult

for LLMs to perform reliably. This is especially the case when comments require using one of our category conflict rules: ChatGPT did not seem to handle category conflicts as we had instructed in the prompt.

### D.3 Binary classification per category

This is the final classification scheme we tried and use in our main work.

**Motivation** Given the difficulty in performing multi-class classification and given that many comments do fit into multiple categories, we decided to implement binary classification per category. This involves prompting for each category on whether each label applies to a comment. This annotation scheme also allows each comment to be labelled with more than one category.

Additionally, in previous attempts, we found that **the label names in the prompt affects the LLM’s response.** For example, renaming the `gratitude` category as “thanks” increases the human-model agreement score because the model would otherwise mark every comment that alludes to a grateful sentiment as `gratitude`. Our binary classification approach eliminates the need for label names in prompts and reduces the priors that LLMs may have on certain label names.

**Setup** As our final prompting strategy, each comment is annotated as a binary classification task per category, i.e., does this category apply to this comment? Note that we have moved from multi-class classification (giving each comment just a single label) to multi-label classification (allowing each comment to get multiple labels). Details on this final prompting strategy can

be found in Section 5.2. However, note that a category’s description in the final version of our prompts may be slightly different from the human description for the category, since after multiple prompt engineering attempts, we found those new descriptions to be better at helping the LLM to detect comments in that category. For example, to help the LLM label short adjective comments like “Marvelous!!!” as general, we had to add the underlined part to the general category’s description in our general prompt: “The comment expresses a general sentiment/adjective about or expresses a general/big-picture opinion about the video’s content and/or about the teaching/professional characteristics of the instructor.”

**Results** Results for this binary classification per category method can be found in Section 6.

## E Prompts

This section details the prompts used for the 0-shot, k-shot and k-shot reasoning prompting strategies. We use these prompts to get the model annotations used in Section 6.

### E.1 0-shot prompts

- Figure 8 is the 0-shot prompt for general.
- Figure 9 is the 0-shot prompt for confusion.
- Figure 10 is the 0-shot prompt for pedagogy.
- Figure 11 is the 0-shot prompt for setup.
- Figure 12 is the 0-shot prompt for personal.
- Figure 13 is the 0-shot prompt for clarification.
- Figure 14 is the 0-shot prompt for gratitude.
- Figure 15 is the 0-shot prompt for nonenglish.
- Figure 16 is the 0-shot prompt for na.

### E.2 k-shot prompts

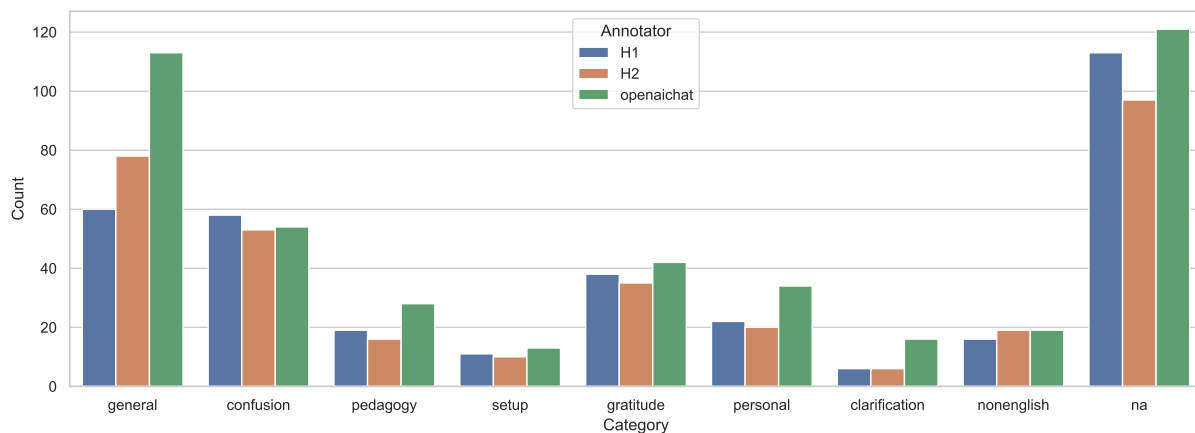
- Figure 17 is the k-shot prompt for general.
- Figure 18 is the k-shot prompt for confusion.
- Figure 19 is the k-shot prompt for pedagogy.
- Figure 20 is the k-shot prompt for setup.
- Figure 21 is the k-shot prompt for personal.

- Figure 22 is the k-shot prompt for clarification.
- Figure 23 is the k-shot prompt for gratitude.
- Figure 24 is the k-shot prompt for nonenglish.
- Figure 25 is the k-shot prompt for na.

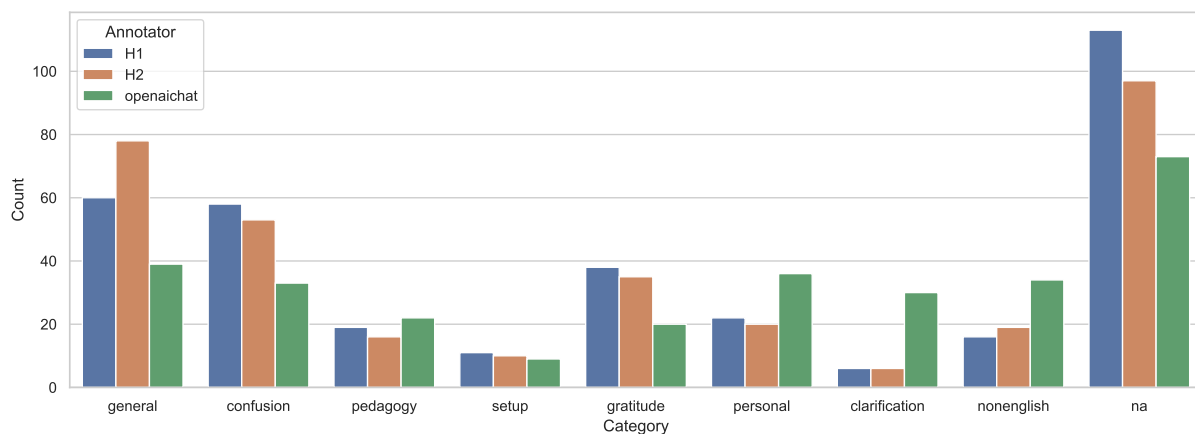
### E.3 k-shot reasoning prompts

- Figure 26 is the k-shot reasoning prompt for general.
- Figure 27 is the k-shot reasoning prompt for confusion.
- Figure 28 is the k-shot reasoning prompt for pedagogy.
- Figure 29 is the k-shot reasoning prompt for setup.
- Figure 30 is the k-shot reasoning prompt for personal.
- Figure 31 is the k-shot reasoning prompt for clarification.
- Figure 32 is the k-shot reasoning prompt for gratitude.
- Figure 33 is the k-shot reasoning prompt for nonenglish.
- Figure 34 is the k-shot reasoning prompt for na.

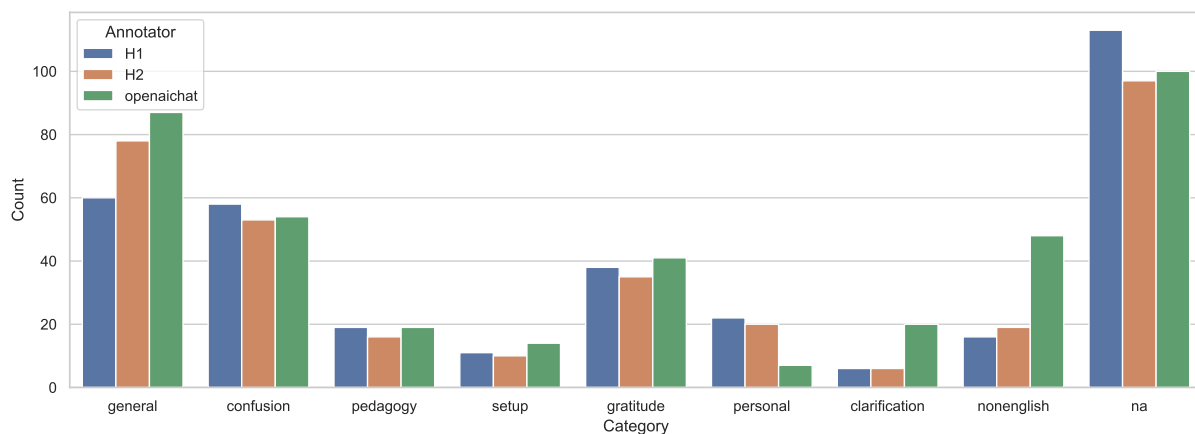




(a) Zero-shot

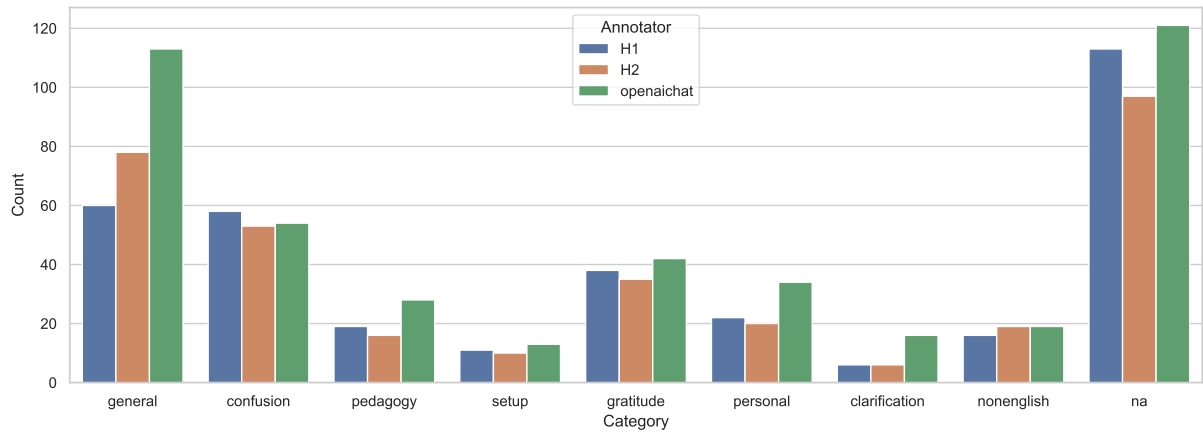


(b) K-shot



(c) K-shot with reasoning

Figure 4: The distribution of categories annotated by the two humans H1 and H2 as well as (a) the 0-shot prompted model, (b) the k-shot prompted model, and (c) the k-shot with reasoning prompted model.



### Annotating YT comments

Instructions  
Current progress: 1% completed, 1 / 80

**Playlist:** MIT 18.06 Linear Algebra, Spring 2005  
**Video:** 11. Matrix Spaces; Rank 1; Small World Graphs  
**Comment:** 06:58 Anybody knows that why the dimension of symmetric matrix and upper triangular matrix are 6?

Do the following annotation labels apply?

**General:** The comment gives a general adjective about or expresses a "general/big-picture" opinion about the video's "content" and/or about the teaching/professional characteristics of the instructor.  
 Yes  No

**Teaching setup:** The comment describes or mentions the lecture's teaching setup. Teaching setup includes the chalk, chalkboard, mic or audio-related aspects, and camera or camera-related aspects (eg. angle).  
 Yes  No

**Pedagogy:** The comment mentions the teacher's instructional method. Instructional methods include the use of examples, applications, worked out problems, proofs, visualizations, elaboration, and analogies.  
 Yes  No

**Confusion:** The comment asks a "math-related" question, expresses "math-related" confusion, and/or points out a "math-related" mistake in the video.  
 Yes  No

**Gratitude:** The comment contains the word "thanks" or "thank you" or "thank".  
 Yes  No

**Personal experience:** The comment mentions the user's personal experience or context with respect to the lecture. Personal experience or context includes the user's own math learning or teaching experiences.  
 Yes  No

**Clarification:** The comment clarifies someone's "math-related" misunderstanding or elaborates content from the video, and the comment includes an "@" that is immediately followed by a username.  
 Yes  No

**Bookmark:** The comment uses timestamps to mark big-picture mathematical concepts in the video.  
 Yes  No

**Non-English:** The comment is not in English.  
 Yes  No

**N/A:** The comment expresses a joke or is a troll comment, and/or the comment says something that is hard to connect to the video content, and/or the comment does not fall into any of the categories above.  
 Yes  No

CONTINUE

Figure 5: Human annotation interface for labelling YouTube comments

## Annotating YT comments

Instructions  
Current progress: 1 % completed, 1 / 80

Playlist: MIT 18.06 Linear Algebra, Spring 2005

Video: 11. Matrix Spaces; Rank 1; Small World Graphs

Comment: 06:58 Anybody knows that why the dimension of symmetric matrix and upper triangular matrix are 6?

Do the following annotation labels apply?

**General:** The comment gives a general adjective about or expresses a "general/big-picture" opinion about the video's "content" and/or about the teaching/professional characteristics of the instructor.

Yes  No

**Teaching setup:** The comment describes or mentions the lecture's teaching setup. Teaching setup includes the chalk, chalkboard, mic or audio-related aspects, and camera or camera-related aspects (eg. angle).

Yes  No

**Pedagogy:** The comment mentions the teacher's instructional method. Instructional methods include the use of examples, applications, worked out problems, proofs, visualizations, elaboration, and analogies.

Yes  No

**Confusion:** The comment asks a "math-related" question, expresses "math-related" confusion, and/or points out a "math-related" mistake in the video.

Yes  No

**Gratitude:** The comment contains the word "thanks" or "thank you" or "thank".

Yes  No

**Personal experience:** The comment mentions the user's personal experience or context with respect to the lecture. Personal experience or context includes the user's own math learning or teaching experiences.

Yes  No

**Clarification:** The comment clarifies someone's "math-related" misunderstanding or elaborates content from the video, and the comment includes an '@' that is immediately followed by a username.

Yes  No

**Bookmark:** The comment uses timestamps to mark big-picture mathematical concepts in the video.

Yes  No

**Non-English:** The comment is not in English.

Yes  No

**N/A:** The comment expresses a joke or is a troll comment, and/or the comment says something that is hard to connect to the video content, and/or the comment does not fall into any of the categories above.

Yes  No

CONTINUE

Figure 6: Human annotation interface for labelling YouTube comments

## Multi-class classification with entire rubric

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: {playlistName}

Video name: {videoName}

Your task is to annotate the comment with one of the labels defined below. If the comment fits into multiple labels, please choose the label that best fits the spirit of the comment.

- 'thanks': The student explicitly expresses gratitude. Comments must include "thanks" or "thank" to be labeled as 'thanks'.
- 'general': The student makes a general comment about the video's \*content\* or a comment about the \*teaching characteristics\* of the instructor. If the comment is not related to the content or teaching characteristics (e.g., the comment is about the instructor's appearance or accent), then the comment should be labeled as 'na'.
- 'style': The student comments on \*how\* the teacher teaches the content. This includes comments on the use of examples, applications, or step-by-step explanations.
- 'personal\_experience': The student shares a personal experience related to the content, such as their previous attempts at learning the content.
- 'question': The student expresses \*math-related\* confusion. Comments must include a question and math-related confusion to be labeled as 'question'.
- 'assist': The student clarifies someone's \*math-related\* misunderstanding or elaborates the content. Comments must include '@user and math-related content to be labeled as 'assist'.
- 'bookmark': The student uses timestamps to mark \*content-related\* features of the video, such as the content outline or the start of a topic. Comments must include a timestamp and math-related content to be labeled as 'bookmark'. Otherwise (e.g., if a student uses a timestamp to mark a joke), the comment should be labeled as 'na'.
- 'non\_english': The student's comment is not in English.
- 'na': The student does not express any of the above types of comments. Instead, for example, the student jokes, says something insulting about the content or instructor, or says something unrelated to the math content.

Comment: {comment}

Label:

Figure 7: Example of a multi-class classification prompt with entire rubric.

#### Zero-shot prompting for general category

Consider a YouTube comment from the math MIT OCW video below:  
Playlist name: {playlistName}  
Video name: {videoName}  
Comment: {comment}

If the statement below is true, please respond "true"; otherwise, please respond "false":  
The comment expresses a general sentiment/adjective about or expresses a \*general/big-picture\* opinion about the video's \*content\* and/or about the teaching/professional characteristics of the \*instructor\*.

Figure 8: The zero-shot prompt for the general category.

#### Zero-shot prompting for confusion category

Consider a YouTube comment from the math MIT OCW video below:  
Playlist name: {playlistName}  
Video name: {videoName}  
Comment: {comment}

If the statement below is true, please respond "true"; otherwise, please respond "false":  
The comment asks a specific mathematical question and/or points out a mathematical mistake in the video.

Figure 9: The zero-shot prompt for the confusion category.

#### Zero-shot prompting for pedagogy category

Consider a YouTube comment from the math MIT OCW video below:  
Playlist name: {playlistName}  
Video name: {videoName}  
Comment: {comment}

If the statement below is true, please respond "true"; otherwise, please respond "false":  
The comment mentions the teacher's instructional method, which includes but is not limited to the use of examples, applications, worked out problems, proofs, visualizations, elaboration, and analogies.

Figure 10: The zero-shot prompt for the pedagogy category.

#### Zero-shot prompting for setup category

Consider a YouTube comment from the math MIT OCW video below:  
Playlist name: {playlistName}  
Video name: {videoName}  
Comment: {comment}

If the statement below is true, please respond "true"; otherwise, please respond "false":  
The comment mentions the lecture's physical teaching setup, which includes but is not limited to the chalk, board, microphone or audio-related aspects, and camera-related aspects (e.g., angle).

Figure 11: The zero-shot prompt for the setup category.

#### Zero-shot prompting for personal category

Consider a YouTube comment from the math MIT OCW video below:  
Playlist name: {playlistName}  
Video name: {videoName}  
Comment: {comment}

If the statement below is true, please respond "true"; otherwise, please respond "false":  
The comment mentions the user's personal experience learning or teaching math on their own outside of watching this lecture/series.

Figure 12: The zero-shot prompt for the personal category.

#### Zero-shot prompting for clarification category

Consider a YouTube comment from the math MIT OCW video below:  
Playlist name: {playlistName}  
Video name: {videoName}  
Comment: {comment}

If the statement below is true, please respond "true"; otherwise, please respond "false":  
The comment clarifies someone's *math-related* misunderstanding or elaborates content from the video, and the comment includes an '@' that is immediately followed by a username.

Figure 13: The zero-shot prompt for the clarification category.

#### Zero-shot prompting for gratitude category

Consider a YouTube comment from the math MIT OCW video below:  
Playlist name: {playlistName}  
Video name: {videoName}  
Comment: {comment}

If the statement below is true, please respond "true"; otherwise, please respond "false":  
The comment contains the word "thanks" or "thank".

Figure 14: The zero-shot prompt for the gratitude category.

#### Zero-shot prompting for nonenglish category

Consider a YouTube comment from the math MIT OCW video below:  
Playlist name: {playlistName}  
Video name: {videoName}  
Comment: {comment}

If the statement below is true, please respond "true"; otherwise, please respond "false":  
The comment is in English.

Figure 15: The zero-shot prompt for the nonenglish category. The final label on this is flipped.

#### Zero-shot prompting for na category

Consider a YouTube comment from the math MIT OCW video below:  
Playlist name: {playlistName}  
Video name: {videoName}  
Comment: {comment}

If the statement below is true, please respond "true"; otherwise, please respond "false":  
The comment expresses a joke or is a troll comment.

Figure 16: The zero-shot prompt for the na category.

### K-shot prompting for general category

Given a user comment on YouTube from a math MIT OCW video, your task is to label whether the comment expresses a general sentiment/adjective about or expresses a *general/big-picture* opinion about the video's *content* and/or about the teaching/professional characteristics of the *instructor*. If it is true, then label "true"; otherwise, label "false".

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.06 Linear Algebra, Spring 2005

Video name: 34. Final Course Review

Comment: Absolutely well done and definitely keep it up!!! :thumbs\_up::thumbs\_up:

:thumbs\_up::thumbs\_up::thumbs\_up::thumbs\_up::thumbs\_up::thumbs\_up:

:thumbs\_up::thumbs\_up::thumbs\_up::thumbs\_up::thumbs\_up::thumbs\_up:

Task: Does the comment express a general opinion about the video's content and/or about the teaching/professional characteristics of the instructor?

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.02 Multivariable Calculus, Fall 2007

Video name: Lec 3: Matrices; inverse matrices | MIT 18.02 Multivariable Calculus, Fall 2007

Comment: Ideally, do you learn multivariable calculus first or linear algebra? A lot of stuff here seems to be based on 18.06.

Task: Does the comment express a general opinion about the video's content and/or about the teaching/professional characteristics of the instructor?

Label: false

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.02 Multivariable Calculus, Fall 2007

Video name: Lec 16: Double integrals | MIT 18.02 Multivariable Calculus, Fall 2007

Comment: This video is very helpful, i appreciate the help.

Task: Does the comment express a general opinion about the video's content and/or about the teaching/professional characteristics of the instructor?

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: {playlistName}

Video name: {videoName}

Comment: {comment}

Task: Does the comment express a general opinion about the video's content and/or about the teaching/professional characteristics of the instructor?

Label:

Figure 17: The k-shot prompt for the general category.

### K-shot prompting for confusion category

Given a user comment on YouTube from a math MIT OCW video, your task is to label whether the comment asks a specific mathematical question and/or points out a mathematical mistake in the video. If it is true, then label "true"; otherwise, label "false".

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.01 Single Variable Calculus, Fall 2006 Video name: Lec 35 | MIT 18.01 Single Variable Calculus, Fall 2007

Comment: can't L'Hopital's rule be explained geometricly? what about the functions curves' tangency ?

Task: Does the comment ask a specific mathematical question and/or points out a mathematical mistake in the video?

Label: true

Consider a YouTube comment from the math MIT OCW video below: Playlist name: MIT 18.06 Linear Algebra, Spring 2005

Video name: 14. Orthogonal Vectors and Subspaces

Comment: Just I have wondered. Are they student of MIT? Why are they so silent??????

Task: Does the comment ask a specific mathematical question and/or points out a mathematical mistake in the video?

Label: false

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.100A Real Analysis, Fall 2020

Video name: Lecture 7: Convergent Sequences of Real Numbers

Comment: There is a mistake in lecture notes, example 71. Example in the lecture notes picks  $\epsilon_0=12$  and then proceeds with  $1=|(-1)^M-(-1)^{(M+1)}|$ . This is wrong. Epsilon should be 1; and the expression with absolute values evaluates to 2. The lecture video is correct, the lecture notes are not.

Task: Does the comment ask a specific mathematical question and/or points out a mathematical mistake in the video?

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: {playlistName}

Video name: {videoName}

Comment: {comment}

Task: Does the comment ask a specific mathematical question and/or points out a mathematical mistake in the video?

Label

Figure 18: The k-shot prompt for the confusion category.

### K-shot prompting for pedagogy category

Given a user comment on YouTube from a math MIT OCW video, your task is to label whether the comment explicitly mentions a pedagogical method, which includes but is not limited to the use of examples, applications, worked out problems, proofs, visualizations, elaboration, step-by-step explanation, reiteration, and analogies. If this is true, then label "true"; otherwise, label "false".

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.06 Linear Algebra, Spring 2005

Video name: 26. Complex Matrices; Fast Fourier Transform

Comment: He's just showing applications of linear algebra, not teaching them. That's why it seems "sloppy". You just can't teach Fourier Transform in 30 mins.

Task: Does the comment explicitly mention a pedagogical method?

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.06 Linear Algebra, Spring 2005

Video name: 1. The Geometry of Linear Equations

Comment: This lecture plus 3blue1brown's videos are getting these concepts to stick for me. Thank you Prof. Strang!!!

Task: Does the comment explicitly mention a pedagogical method?

Label: false

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.06 Linear Algebra, Spring 2005

Video name: 9. Independence, Basis, and Dimension

Comment: His teaching style seems casual and intuitive. I go to a small public college and the course is much more formal and proof driven. These lectures are a great addition to (as well as a nice break from) formal proofs. Thanks MIT!

Task: Does the comment explicitly mention a pedagogical method?

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: {playlistName}

Video name: {videoName}

Comment: {comment}

Task: Does the comment explicitly mention a pedagogical method?

Label:

Figure 19: The k-shot prompt for the pedagogy category.



### K-shot prompting for setup category

Given a user comment on YouTube from a math MIT OCW video, your task is to label whether the comment mentions the lecture's physical teaching setup, which includes but is not limited to the chalk, board, microphone or audio-related aspects, and camera-related aspects (e.g., angle). If it is true, then label "true"; otherwise, label "false".

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.100A Real Analysis, Fall 2020

Video name: Lecture 1: Sets, Set Operations and Mathematical Induction

Comment: Thanks for posting this course, the instructor is great. If I may, there is only one request, in the future if the camera could move less frequently, the camera is following the instructor too closely, making me a bit dizzy.

Task: Does the comment mention the lecture's physical teaching setup?

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: 6.041 Probabilistic Systems Analysis and Applied Probability

Video name: 5. Discrete Random Variables I

Comment: A "random variable is a function in programming"... mic drop!

Task: Does the comment mention the lecture's physical teaching setup?

Label: false

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.01 Single Variable Calculus, Fall 2006

Video name: Lec 30 | MIT 18.01 Single Variable Calculus, Fall 2007

Comment: The mic noise and hiss is distracting in this lecture, I hope someone could fix it ..

Task: Does the comment mention the lecture's physical teaching setup?

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: {playlistName}

Video name: {videoName}

Comment: {comment}

Task: Does the comment mention the lecture's physical teaching setup?

Label:

Figure 20: The k-shot prompt for the setup category.

### K-shot prompting for personal category

Given a user comment on YouTube from a math MIT OCW video, your task is to label whether the comment mentions the user's personal experience learning or teaching math on their own outside of watching this lecture/series. If it is true, then label "true"; otherwise, label "false".

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.06 Linear Algebra, Spring 2005

Video name: 1. The Geometry of Linear Equations

Comment: Amazing! I like linear algebra a lot, I already had this class in college, I keep reading about it and ... I didn't even notice the passing of 40 minutes of the first class you! No wonder MIT is a world reference!

Task: Does the comment mention the user's personal experience learning or teaching math on their own outside of watching this lecture/series?

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: 6.041 Probabilistic Systems Analysis and Applied Probability

Video name: 14. Poisson Process I

Comment: I am having a hard time making sense of the notation at 11:22. I believe the notation should be the conditional probability  $P(k|t)$  rather than  $P(k,t)$ . I interpreted the latter to be the joint probability and if it is the case, the summation over all  $k$  of  $P(k,t)$  given a fixed  $t$  could not be equal to 1. Anyone, please help knock some sense to my head!

Task: Does the comment mention the user's personal experience learning or teaching math on their own outside of watching this lecture/series?

Label: false

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.06 Linear Algebra, Spring 2005

Video name: 21. Eigenvalues and Eigenvectors

Comment: Wish this guy taught me Math 293 and 294 at Cornell. My guy could barely speak English, let alone explain what we were trying to accomplish. I understood that if we wanted eigenvectors perpendicular to  $x$  we'd get lift relative to flow...but this guy would have made the math a bit simpler.

Task: Does the comment mention the user's personal experience learning or teaching math on their own outside of watching this lecture/series?

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: {playlistName}

Video name: {videoName}

Comment: {comment}

Task: Does the comment mention the user's personal experience learning or teaching math on their own outside of watching this lecture/series?

Label:

Figure 21: The k-shot prompt for the personal category.

### K-shot prompting for clarification category

Given a user comment on YouTube from a math MIT OCW video, your task is to label whether the comment clarifies someone's *\*math-related\** misunderstanding or elaborates content from the video, and the comment includes an '@' that is immediately followed by a username. If this is true, then label "true"; otherwise, label "false".

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.01 Single Variable Calculus, Fall 2006

Video name: Lec 3 | MIT 18.01 Single Variable Calculus, Fall 2007

Comment: @[USERNAME] it's the math dragon theorem

Task: Does the comment clarify someone's *\*math-related\** misunderstanding or elaborate content from the video?

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.02 Multivariable Calculus, Fall 2007

Video name: Lec 23: Flux; normal form of Green's theorem | MIT 18.02 Multivariable Calculus, Fall 2007

Comment: 30:00, the way to remember it is that the work is a straightforward dot product of  $F$  with  $\langle dx, dy \rangle$ ,  $M$  goes with  $x$  and  $N$  goes with  $y$  and we add, and the flux is a dot product of  $F$  with the same vector rotated  $\pi/2$  so  $N$  goes with  $x$  and a minus sign with few choices left for  $M$ . Auroux missed a nice opportunity at the beginning to clarify the sign convention for flux by foreshadowing the result for closed curves with  $+$  being from the inside, out. I'm not faulting anyone, I couldn't give a lecture on this and keep possession of both my hands when erasing blackboards operated by hazardous machines. If he loses his hands, he'll never erase anything again. Be careful out there, Denis, we don't want to lose a great teacher. Task: Does the comment clarify someone's *\*math-related\** misunderstanding or elaborate content from the video?

Label: false

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.01 Single Variable Calculus, Fall 2006

Video name: Lec 22 | MIT 18.01 Single Variable Calculus, Fall 2007

Comment: @[USERNAME] Actually, if a constant  $k=11m$  is used, then in the final formula for  $V$  you will end up with subtracting  $m^1$  from  $m^2$  which is apparently not correct.

Task: Does the comment clarify someone's *\*math-related\** misunderstanding or elaborate content from the video?

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: {playlistName}

Video name: {videoName}

Comment: {comment}

Task: Does the comment clarify someone's *\*math-related\** misunderstanding or elaborate content from the video?

Label:

Figure 22: The k-shot prompt for the clarification category.

### K-shot prompting for gratitude category

Given a user comment on YouTube from a math MIT OCW video, your task is to label whether the comment contains the word "thanks" or "thank". If it is true, then label "true"; otherwise, label "false".

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.650 Statistics for Applications, Fall 2016

Video name: 15. Regression (cont.)

Comment: Thank you for the lectures, could you please state what topics did Lectures 10 and 16 covered? So we can research them separately.

Task: Does the comment contains the word "thanks" or "thank"?

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.100A Real Analysis, Fall 2020

Video name: Lecture 1: Sets, Set Operations and Mathematical Induction

Comment: "Keep up the good work:thumbs\_up::thumbs\_up:

Task: Does the comment contains the word "thanks" or "thank"?

Label: false

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.01 Single Variable Calculus, Fall 2006

Video name: Lec 2 | MIT 18.01 Single Variable Calculus, Fall 2007

Comment: Thanks! I prepared my high school final exam from this lecture. This really helped me!!

Task: Does the comment contains the word "thanks" or "thank"?

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: {playlistName}

Video name: {videoName}

Comment: {comment}

Task: Does the comment contains the word "thanks" or "thank"?

Label:

Figure 23: The k-shot prompt for the gratitude category.

### K-shot prompting for nonenglish category

Given a user comment on YouTube from a math MIT OCW video, your task is to label whether the comment is in English. If it is true, then label "true"; otherwise, label "false".

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.06 Linear Algebra, Spring 2005

Video name: 1. The Geometry of Linear Equations

Comment: Amazing! I like linear algebra a lot, I already had this class in college, I keep reading about it and ... I didn't even notice the passing of 40 minutes of the first class you! No wonder MIT is a world reference!

Task: Is the comment in English?

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.01 Single Variable Calculus, Fall 2006

Video name: Lec 35 | MIT 18.01 Single Variable Calculus, Fall 2007

Comment: 이게 계속 쓰지 말라던로피탈이구나

Task: Is the comment in English?

Label: false

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.06 Linear Algebra, Spring 2005

Video name: 21. Eigenvalues and Eigenvectors

Comment: Wish this guy taught me Math 293 and 294 at Cornell. My guy could barely speak English, let alone explain what we were trying to accomplish. I understood that if we wanted eigenvectors perpendicular to  $x$  we'd get lift relative to flow...but this guy would have made the math a bit simpler.

Task: Is the comment in English?

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: {playlistName}

Video name: {videoName}

Comment: {comment}

Task: Is the comment in English?

Label:

Figure 24: The k-shot prompt for the nonenglish category.

### K-shot prompting for na category

Given a user comment on YouTube from a math MIT OCW video, your task is to label whether the comment expresses a joke or is a troll comment. If it is true, then label "true"; otherwise, label "false".

Consider a YouTube comment from the math MIT OCW video below:  
Playlist name: MIT 18.02 Multivariable Calculus, Fall 2007  
Video name: Lec 1: Dot product | MIT 18.02 Multivariable Calculus, Fall 2007  
Comment: Watching this to make me feel better about college algebra. lol  
Task: Does the comment expresses a joke or is the comment a troll comment?  
Label: true

Consider a YouTube comment from the math MIT OCW video below:  
Playlist name: MIT 18.06 Linear Algebra, Spring 2005  
Video name: 3. Multiplication and Inverse Matrices  
Comment: oh sir thank you a lot !!!!  
Task: Does the comment expresses a joke or is the comment a troll comment?  
Label: false

Consider a YouTube comment from the math MIT OCW video below:  
Playlist name: MIT 18.02 Multivariable Calculus, Fall 2007  
Video name: Lec 24: Simply connected regions; review | MIT 18.02 Multivariable Calculus, Fall 2007  
Comment: i couldnt resist xD  
Task: Does the comment expresses a joke or is the comment a troll comment?  
Label: true

Consider a YouTube comment from the math MIT OCW video below:  
Playlist name: {playlistName}  
Video name: {videoName}  
Comment: {comment}  
Task: Does the comment expresses a joke or is the comment a troll comment?  
Label:

Figure 25: The k-shot prompt for the na category.

### K-shot reasoning prompting for general category

Given a user comment on YouTube from a math MIT OCW video, your task is to explain (after "Explanation:") and label (after "Label:") whether the comment expresses a general sentiment/adjective about or expresses a \*general/big-picture\* opinion about the video's \*content\* and/or about the teaching/professional characteristics of the \*instructor\*. If it is true, then label "true"; otherwise, label "false".

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.06 Linear Algebra, Spring 2005

Video name: 34. Final Course Review

Comment: Absolutely well done and definitely keep it up!!! :thumbs\_up::thumbs\_up:

:thumbs\_up::thumbs\_up::thumbs\_up::thumbs\_up::thumbs\_up::thumbs\_up:

:thumbs\_up::thumbs\_up::thumbs\_up::thumbs\_up::thumbs\_up::thumbs\_up:

Task: Does the comment express a general opinion about the video's content and/or about the teaching/professional characteristics of the instructor?

Explanation: Let's go through the sentences one by one until we find one

that meets the criterion. "Absolutely well done and definitely keep it up!!!

:thumbs\_up::thumbs\_up::thumbs\_up::thumbs\_up::thumbs\_up::thumbs\_up::thumbs\_up::thumbs\_up:

:thumbs\_up::thumbs\_up::thumbs\_up::thumbs\_up::thumbs\_up::thumbs\_up:" expresses a general opinion about the video (well done). Therefore, the label is true.

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.02 Multivariable Calculus, Fall 2007

Video name: Lec 3: Matrices; inverse matrices | MIT 18.02 Multivariable Calculus, Fall 2007

Comment: Ideally, do you learn multivariable calculus first or linear algebra? A lot of stuff here seems to be based on 18.06.

Task: Does the comment express a general opinion about the video's content and/or about the teaching/professional characteristics of the instructor?

Explanation: Let's go through the sentences one by one until we find one that meets the

criterion. "Ideally, do you learn multivariable calculus first or linear algebra?" asks

a math-related question, and does not express a general opinion about the content or

teaching of the instructor. "A lot of stuff here seems to be based on 18.06." builds on

the math-related question. Therefore, the label is false.

Label: false

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.02 Multivariable Calculus, Fall 2007

Video name: Lec 16: Double integrals | MIT 18.02 Multivariable Calculus, Fall 2007

Comment: This video is very helpful, i appreciate the help.

Task: Does the comment express a general opinion about the video's content and/or about the teaching/professional characteristics of the instructor?

Explanation: Let's go through the sentences one by one until we find one that meets

the criterion. "This video is very helpful, i appreciate the help." expresses a general

opinion of the video (helpful). Therefore, the label is true.

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: {playlistName}

Video name: {videoName}

Comment: {comment}

Task: Does the comment express a general opinion about the video's content and/or about the teaching/professional characteristics of the instructor?

Explanation:

Figure 26: The k-shot reasoning prompt for the general category.

### K-shot reasoning prompting for confusion category

Given a user comment on YouTube from a math MIT OCW video, your task is to explain (after "Explanation:") and label (after "Label:") whether the comment asks a specific mathematical question and/or points out a mathematical mistake in the video. If it is true, then label "true"; otherwise, label "false".

Consider a YouTube comment from the math MIT OCW video below:  
Playlist name: MIT 18.01 Single Variable Calculus, Fall 2006  
Video name: Lec 35 | MIT 18.01 Single Variable Calculus, Fall 2007  
Comment: can't L'Hopital's rule be explained geometricly? what about the functions curves' tangency ?  
Task: Does the comment ask a specific mathematical question and/or points out a mathematical mistake in the video?  
Explanation: Let's go through the sentences one by one until we find one that meets the criterion. "can't L'Hopital's rule be explained geometricly?" asks a question about L'Hopital's rule which is an important mathematical concept in calculus. Therefore, the label is true.  
Label: true

Consider a YouTube comment from the math MIT OCW video below:  
Playlist name: MIT 18.06 Linear Algebra, Spring 2005  
Video name: 14. Orthogonal Vectors and Subspaces  
Comment: Just I have wondered. Are they student of MIT? Why are they so silent?????  
Task: Does the comment ask a specific mathematical question and/or points out a mathematical mistake in the video?  
Explanation: Let's go through the sentences one by one until we find one that meets the criterion. "Just I have wondered." is not a question and does not point out a mistake in the video. "Are they student of MIT? Why are they so silent?????" are questions, but it is not related to mathematics. Therefore, the label is false.  
Label: false

Consider a YouTube comment from the math MIT OCW video below:  
Playlist name: MIT 18.100A Real Analysis, Fall 2020  
Video name: Lecture 7: Convergent Sequences of Real Numbers  
Comment: There is a mistake in lecture notes, example 71. Example in the lecture notes picks  $\epsilon_0=1/2$  and then proceeds with  $1=|(-1)^M - (-1)^{(M+1)}|$ . This is wrong. Epsilon should be 1; and the expression with absolute values evaluates to 2. The lecture video is correct, the lecture notes are not.  
Task: Does the comment ask a specific mathematical question and/or points out a mathematical mistake in the video? Explanation: Let's go through the sentences one by one until we find one that meets the criterion. "There is a mistake in lecture notes, example 71." points out a mistake in the lecture notes. Therefore, the label is true.  
Label: true

Consider a YouTube comment from the math MIT OCW video below:  
Playlist name: {playlistName}  
Video name: {videoName}  
Comment: {comment}  
Task: Does the comment ask a specific mathematical question and/or points out a mathematical mistake in the video?  
Explanation:

Figure 27: The k-shot reasoning prompt for the confusion category.



### K-shot reasoning prompting for pedagogy category

Given a user comment on YouTube from a math MIT OCW video, your task is to explain (after "Explanation:") and label (after "Label:") whether the comment explicitly mentions a pedagogical method, which includes but is not limited to the use of examples, applications, worked out problems, proofs, visualizations, elaboration, step-by-step explanation, reiteration, and analogies. If this is true, then label "true"; otherwise, label "false".

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.06 Linear Algebra, Spring 2005

Video name: 26. Complex Matrices; Fast Fourier Transform

Comment: He's just showing applications of linear algebra, not teaching them. That's why it seems "sloppy". You just can't teach Fourier Transform in 30 mins.

Task: Does the comment explicitly mention a pedagogical method?

Explanation: Let's go through the sentences one by one until we find one that meets the criterion. "He's just showing applications of linear algebra, not teaching them." mentions the teacher is using applications. Applications are a pedagogical methods. Therefore, the label is true.

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.06 Linear Algebra, Spring 2005

Video name: 1. The Geometry of Linear Equations

Comment: This lecture plus 3blue1brown's videos are getting these concepts to stick for me. Thank you Prof. Strang!!! Task: Does the comment explicitly mention a pedagogical method?

Explanation: Let's go through the sentences one by one until we find one that meets the criterion. "This lecture plus 3blue1brown's videos are getting these concepts to stick for me." communicates that the video is helpful, but it does not mention any pedagogical method that makes the video helpful. "Thank you Prof. Strang!!!" does not mention any pedagogical method. Therefore, the label is false.

Label: false

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.06 Linear Algebra, Spring 2005

Video name: 9. Independence, Basis, and Dimension

Comment: His teaching style seems casual and intuitive. I go to a small public college and the course is much more formal and proof driven. These lectures are a great addition to (as well as a nice break from) formal proofs. Thanks MIT!

Task: Does the comment explicitly mention a pedagogical method?

Explanation: Let's go through the sentences one by one until we find one that meets the criterion. "His teaching style seems casual and intuitive." describes the teaching style, but does not mention what methods the instructor uses to enable for a casual and intuitive style. "I go to a small public college and the course is much more formal and proof driven." mentions the proofs from their previous course, which is a pedagogical method. Therefore, the label is true.

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: {playlistName}

Video name: {videoName}

Comment: {comment}

Task: Does the comment explicitly mention a pedagogical method?

Explanation:

Figure 28: The k-shot reasoning prompt for the pedagogy category.

### K-shot reasoning prompting for setup category

Given a user comment on YouTube from a math MIT OCW video, your task is to explain (after "Explanation:") and label (after "Label:") whether the comment mentions the lecture's physical teaching setup, which includes but is not limited to the chalk, board, microphone or audio-related aspects, and camera-related aspects (e.g., angle). If it is true, then label "true"; otherwise, label "false".

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.100A Real Analysis, Fall 2020

Video name: Lecture 1: Sets, Set Operations and Mathematical Induction

Comment: Thanks for posting this course, the instructor is great. If I may, there is only one request, in the future if the camera could move less frequently, the camera is following the instructor too closely, making me a bit dizzy.

Task: Does the comment mention the lecture's physical teaching setup?

Explanation: Let's go through the sentences one by one until we find one that meets the criterion. "Thanks for posting this course, the instructor is great." does not mention the lecture's physical teaching setup. "If I may, there is only one request, in the future if the camera could move less frequently, the camera is following the instructor too closely, making me a bit dizzy." mentions the camera, which is a part of the lecture's physical setup. Therefore, the label is true.

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: 6.041 Probabilistic Systems Analysis and Applied Probability

Video name: 5. Discrete Random Variables I

Comment: A "random variable is a function in programming"... mic drop!

Task: Does the comment mention the lecture's physical teaching setup?

Explanation: Let's go through the sentences one by one until we find one that meets the criterion. "A "random variable is a function in programming"... mic drop!" mentions a mic, but is used figuratively in this context. Therefore, the label is false.

Label: false

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.01 Single Variable Calculus, Fall 2006

Video name: Lec 30 | MIT 18.01 Single Variable Calculus, Fall 2007

Comment: The mic noise and hiss is distracting in this lecture, I hope someone could fix it ..

Task: Does the comment mention the lecture's physical teaching setup?

Explanation: Let's go through the sentences one by one until we find one that meets the criterion. "The mic noise and hiss is distracting in this lecture, I hope someone could fix it .." mentions the mic hissing, which is part of the physical teaching setup. Therefore, the label is true.

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: {playlistName}

Video name: {videoName}

Comment: {comment}

Task: Does the comment mention the lecture's physical teaching setup?

Explanation:

Figure 29: The k-shot reasoning prompt for the setup category.

### K-shot reasoning prompting for personal category

Given a user comment on YouTube from a math MIT OCW video, your task is to explain (after "Explanation:") and label (after "Label:") whether the comment mentions the user's personal experience learning or teaching math on their own outside of watching this lecture/series. If it is true, then label "true"; otherwise, label "false".

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.06 Linear Algebra, Spring 2005

Video name: 1. The Geometry of Linear Equations

Comment: Amazing! I like linear algebra a lot, I already had this class in college, I keep reading about it and ... I didn't even notice the passing of 40 minutes of the first class you! No wonder MIT is a world reference!

Task: Does the comment mention the user's personal experience learning or teaching math on their own outside of watching this lecture/series?

Explanation: Let's go through the sentences one by one until we find one that mentions the user's personal experience. "Amazing!" expresses the user's opinion about the content, but does not mention their personal experience outside of this lecture. "I like linear algebra a lot, I already had this class in college, I keep reading about it and ... I didn't even notice the passing of 40 minutes of the first class you!" mentions taking this class in college, which is a personal experience for this user outside of watching this lecture or series. Therefore, the label is true.

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: 6.041 Probabilistic Systems Analysis and Applied Probability

Video name: 14. Poisson Process I

Comment: I am having a hard time making sense of the notation at 11:22. I believe the notation should be the conditional probability  $P(k|t)$  rather than  $P(k,t)$ . I interpreted the latter to be the joint probability and if it is the case, the summation over all  $k$  of  $P(k,t)$  given a fixed  $t$  could not be equal to 1. Anyone, please help knock some sense to my head!

Task: Does the comment mention the user's personal experience learning or teaching math on their own outside of watching this lecture/series?

Explanation: Let's go through the sentences one by one until we find one that mentions the user's personal experience. "I am having a hard time making sense of the notation at 11:22." expresses the user's confusion with the lecture content, but not an experience outside of watching this lecture or series. "I believe the notation should be the conditional probability  $P(k|t)$  rather than  $P(k,t)$ ." elaborates what the user is confused about with the lecture, but not a personal experience outside of the lecture or series. "I interpreted the latter to be the joint probability and if it is the case, the summation over all  $k$  of  $P(k,t)$  given a fixed  $t$  could not be equal to 1." elaborates what the user misunderstood, but does not communicate a personal experience outside of watching this lecture or series. "Anyone, please help knock some sense to my head!" requests for help from others, but does not talk about a personal experience outside of this lecture or series. Therefore, the label is false.

Label: false

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.06 Linear Algebra, Spring 2005

Video name: 21. Eigenvalues and Eigenvectors

Comment: Wish this guy taught me Math 293 and 294 at Cornell. My guy could barely speak English, let alone explain what we were trying to accomplish. I understood that if we wanted eigenvectors perpendicular to  $x$  we'd get lift relative to flow...but this guy would have made the math a bit simpler.

Task: Does the comment mention the user's personal experience learning or teaching math on their own outside of watching this lecture/series?

Explanation: Let's go through the sentences one by one until we find one that mentions the user's personal experience. "Wish this guy taught me Math 293 and 294 at Cornell." mentions the user's own math classes at a different university. This is a personal experience related to learning outside of this video and lecture series. Therefore, the label is true.

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: {playlistName}

Video name: {videoName}

Comment: {comment}

Task: Does the comment mention the user's personal experience learning or teaching math on their own outside of watching this lecture/series?

Explanation:

### K-shot reasoning prompting for clarification category

Given a user comment on YouTube from a math MIT OCW video, your task is to explain (after "Explanation:") and label (after "Label:") whether the comment clarifies someone's *\*math-related\** misunderstanding or elaborates content from the video, and the comment includes an '@' that is immediately followed by a username. If this is true, then label "true"; otherwise, label "false".

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.01 Single Variable Calculus, Fall 2006

Video name: Lec 3 | MIT 18.01 Single Variable Calculus, Fall 2007

Comment: @[USERNAME] it's the math dragon theorem

Task: Does the comment clarify someone's *\*math-related\** misunderstanding or elaborate content from the video?

Explanation: Let's go through the sentences one by one until we find one that meets the criterion. "[USERNAME] it's the math dragon theorem" tags another user, and seems to respond to a question from this user. Responding to a question is a form of clarification. Therefore, the label is true.

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.02 Multivariable Calculus, Fall 2007

Video name: Lec 23: Flux; normal form of Green's theorem | MIT 18.02 Multivariable Calculus, Fall 2007

Comment: 30:00, the way to remember it is that the work is a straightforward dot product of  $F$  with  $\langle dx, dy \rangle$ ,  $M$  goes with  $x$  and  $N$  goes with  $y$  and we add, and the flux is a dot product of  $F$  with the same vector rotated  $\pi/2$  so  $N$  goes with  $x$  and a minus sign with few choices left for  $M$ . Auroux missed a nice opportunity at the beginning to clarify the sign convention for flux by foreshadowing the result for closed curves with  $+$  being from the inside, out. I'm not faulting anyone, I couldn't give a lecture on this and keep possession of both my hands when erasing blackboards operated by hazardous machines. If he loses his hands, he'll never erase anything again. Be careful out there, Denis, we don't want to lose a great teacher.

Task: Does the comment clarify someone's *\*math-related\** misunderstanding or elaborate content from the video?

Explanation: Let's go through the sentences one by one until we find one that meets the criterion. "30:00, the way to remember it is that the work is a straightforward dot product of  $F$  with  $\langle dx, dy \rangle$ ,  $M$  goes with  $x$  and  $N$  goes with  $y$  and we add, and the flux is a dot product of  $F$  with the same vector rotated  $\pi/2$  so  $N$  goes with  $x$  and a minus sign with few choices left for  $M$ ." does not contain any @ symbol. "Auroux missed a nice opportunity at the beginning to clarify the sign convention for flux by foreshadowing the result for closed curves with  $+$  being from the inside, out." also does not contain the @ symbol. "Also I'm not faulting anyone, I couldn't give a lecture on this and keep possession of both my hands when erasing blackboards operated by hazardous machines." also does not contain the @ symbol. "If he loses his hands, he'll never erase anything again. Be careful out there, Denis, we don't want to lose a great teacher." also does not contain the @ symbol. Therefore the label is false.

Label: false

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.01 Single Variable Calculus, Fall 2006

Video name: Lec 22 | MIT 18.01 Single Variable Calculus, Fall 2007

Comment: @[USERNAME] Actually, if a constant  $k=11m$  is used, then in the final formula for  $V$  you will end up with subtracting  $m^1$  from  $m^2$  which is apparently not correct.

Task: Does the comment clarify someone's *\*math-related\** misunderstanding or elaborate content from the video?

Explanation: Let's go through the sentences one by one until we find one that meets the criterion. "[USERNAME] Actually, if a constant  $k=11m$  is used, then in the final formula for  $V$  you will end up with subtracting  $m^1$  from  $m^2$  which is apparently not correct." contains the @ symbol and seems to correct the other user's understanding of the math formula. Correcting someone's understanding of the math formula is a form of clarification. Therefore, the label is true.

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: {playlistName}

Video name: {videoName}

Comment: {comment}

Task: Does the comment clarify someone's *\*math-related\** misunderstanding or elaborate content from the video?

Explanation:

Figure 31: The k-shot reasoning prompt for the clarification category.

### K-shot reasoning prompting for gratitude category

Given a user comment on YouTube from a math MIT OCW video, your task is to explain (after "Explanation:") and label (after "Label:") whether the comment contains the word "thanks" or "thank". If it is true, then label "true"; otherwise, label "false".

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.650 Statistics for Applications, Fall 2016

Video name: 15. Regression (cont.)

Comment: Thank you for the lectures, could you please state what topics did Lectures 10 and 16 covered? So we can research them separately.

Task: Does the comment contains the word "thanks" or "thank"?

Explanation: Let's go through the sentences one by one until we find one that meets the criterion. "Thank you for the lectures, could you please state what topics did Lectures 10 and 16 covered?" contains one of the expressions. Therefore, the label is true.

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.100A Real Analysis, Fall 2020

Video name: Lecture 1: Sets, Set Operations and Mathematical Induction

Comment: Keep up the good work:thumbs\_up::thumbs\_up:

Task: Does the comment contains the word "thanks" or "thank"?

Explanation: Let's go through the sentences one by one until we find one that meets the criterion. "Keep up the good work:thumbs\_up::thumbs\_up:" does not contain any of the expressions. Therefore, the label is false.

Label: false

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.01 Single Variable Calculus, Fall 2006

Video name: Lec 2 | MIT 18.01 Single Variable Calculus, Fall 2007

Comment: Thanks! I prepared my high school final exam from this lecture. This really helped me!!

Task: Does the comment contains the word "thanks" or "thank"?

Explanation: Let's go through the sentences one by one until we find one that meets the criterion. "Thanks!" contains one of the expressions. Therefore, the label is true.

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: {playlistName}

Video name: {videoName}

Comment: {comment}

Task: Does the comment contains the word "thanks" or "thank"?

Explanation:

Figure 32: The k-shot reasoning prompt for the gratitude category.

### K-shot reasoning prompting for nonenglish category

Given a user comment on YouTube from a math MIT OCW video, your task is to explain (after "Explanation:") and label (after "Label:") whether the comment is in English. If it is true, then label "true"; otherwise, label "false".

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.06 Linear Algebra, Spring 2005

Video name: 1. The Geometry of Linear Equations

Comment: Amazing! I like linear algebra a lot, I already had this class in college, I keep reading about it and ... I didn't even notice the passing of 40 minutes of the first class you! No wonder MIT is a world reference!

Task: Is the comment in English?

Explanation: Let's go through the sentences one by one until we find a sentence that is not in English. "Amazing!" is in English. "I like linear algebra a lot, I already had this class in college, I keep reading about it and ... I didn't even notice the passing of 40 minutes of the first class you!" is in English. "No wonder MIT is a world reference!" is also in English. The entire comment is in English. Therefore, the label is true.

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.01 Single Variable Calculus, Fall 2006

Video name: Lec 35 | MIT 18.01 Single Variable Calculus, Fall 2007

Comment: 이게 계속 쓰지 말라던로 피탈이구나

Task: Is the comment in English?

Explanation: Let's go through the sentences one by one until we find a sentence that is not in English. 이게 계속 쓰지 말라던로 피탈이구나 is not in English. Therefore, the label is false.

Label: false

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.06 Linear Algebra, Spring 2005

Video name: 21. Eigenvalues and Eigenvectors

Comment: Wish this guy taught me Math 293 and 294 at Cornell. My guy could barely speak English, let alone explain what we were trying to accomplish. I understood that if we wanted eigenvectors perpendicular to  $x$  we'd get lift relative to flow...but this guy would have made the math a bit simpler.

Task: Is the comment in English?

Explanation: Let's go through the sentences one by one until we find a sentence that is not in English. "Wish this guy taught me Math 293 and 294 at Cornell." is in English. "My guy could barely speak English, let alone explain what we were trying to accomplish." is in English. "I understood that if we wanted eigenvectors perpendicular to  $x$  we'd get lift relative to flow...but this guy would have made the math a bit simpler." is also in English. The entire comment is in English. Therefore, the label is true.

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: {playlistName}

Video name: {videoName}

Comment: {comment}

Task: Is the comment in English?

Explanation:

Figure 33: The k-shot reasoning prompt for the nonenglish category.

### K-shot reasoning prompting for na category

Given a user comment on YouTube from a math MIT OCW video, your task is to explain (after "Explanation:") and label (after "Label:") whether the comment expresses a joke or is a troll comment. If it is true, then label "true"; otherwise, label "false".

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.02 Multivariable Calculus, Fall 2007

Video name: Lec 1: Dot product | MIT 18.02 Multivariable Calculus, Fall 2007

Comment: Watching this to make me feel better about college algebra. lol

Task: Does the comment expresses a joke or is the comment a troll comment?

Explanation: Let's go through the sentences one by one until we find one that meets the criterion. "Watching this to make me feel better about college algebra." does not seem to express a joke. "lol" expresses a joking tone to the comment. Therefore, the label is true.

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.06 Linear Algebra, Spring 2005

Video name: 3. Multiplication and Inverse Matrices

Comment: oh sir thank you a lot !!!!

Task: Does the comment expresses a joke or is the comment a troll comment?

Explanation: Let's go through the sentences one by one until we find one that meets the criterion. "oh sir thank you a lot !!!!" does not express a joke or troll comment. Therefore, the label is false.

Label: false

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: MIT 18.02 Multivariable Calculus, Fall 2007

Video name: Lec 24: Simply connected regions; review | MIT 18.02 Multivariable Calculus, Fall 2007

Comment: i couldnt resist xD

Task: Does the comment expresses a joke or is the comment a troll comment?

Explanation: Let's go through the sentences one by one until we find one that meets the criterion. "i couldnt resist xD" ends in a joking emoji and expressing a joking tone. Therefore, the label is true.

Label: true

Consider a YouTube comment from the math MIT OCW video below:

Playlist name: {playlistName}

Video name: {videoName}

Comment: {comment}

Task: Does the comment expresses a joke or is the comment a troll comment?

Explanation:

Figure 34: The k-shot reasoning prompt for the na category.

# Recognizing Learner Handwriting Retaining Orthographic Errors for Enabling Fine-Grained Error Feedback

Christian Gold<sup>1</sup>, Ronja Laarmann-Quante<sup>2</sup> and Torsten Zesch<sup>1</sup>

<sup>1</sup>CATALPA, FernUniversität in Hagen, Germany,

<sup>2</sup>Ruhr University Bochum, Faculty of Philology, Department of Linguistics, Germany

## Abstract

This paper addresses the problem of providing automatic feedback on orthographic errors in handwritten text. Despite the availability of automatic error detection systems, the practical problem of digitizing the handwriting remains. Current handwriting recognition (HWR) systems produce highly accurate transcriptions but normalize away the very errors that are essential for providing useful feedback, e.g. orthographic errors. Our contribution is twofold: First, we create a comprehensive dataset of handwritten text with transcripts retaining orthographic errors by transcribing 1,350 pages from the German learner dataset FD-LEX. Second, we train a simple HWR system on our dataset, allowing it to transcribe words with orthographic errors. Thereby, we evaluate the effect of different dictionaries on recognition output, highlighting the importance of addressing spelling errors in these dictionaries.

## 1 Introduction

Early L1 learners typically write by hand, even in the digital age, and handwriting remains important (Ray et al., 2022; Danna et al., 2022; Mathwin et al., 2022). Automatic feedback on error types in learner language is available (Laarmann-Quante, 2017; Berkling and Lavalley, 2015), but faces the practical problem of having to digitize the handwriting first. Current *handwriting recognition* (HWR) systems yield very good results (Kizilirmak and Yanikoglu, 2022; Xiao et al., 2020; Li et al., 2021) with one crucial problem: they typically normalize away the orthographic errors (Neto et al., 2020) that are important for giving useful feedback to learners. In Figure 1, when humans read this handwritten word, they look at the shapes of the letters to form hypotheses. The first letter(s) could be a *d* or a *cl* and we decide about this informed by a hypothesis about the whole word. In this case, we see that it is probably supposed to be *dounut*, so the first letter is a *d*. We see that there is an extra letter *u* at

the third position which we ignore for forming our hypothesis about the word, but still recognize so that we could give a learner appropriate feedback about it.

Automatic handwriting recognition systems are typically trained and evaluated on handwritten text along with transcripts that do not contain orthographic errors. Many HWR systems contain a language model component (Scheidl et al., 2018) that is used to further normalize the output. As a result, HWR systems yield ‘clean’ transcripts without any orthographic errors (right branch in Figure 1) that cannot be used to give feedback on orthographic errors. Instead, we need HWR systems outputting transcripts that retain orthographic errors (middle branch in Figure 1).

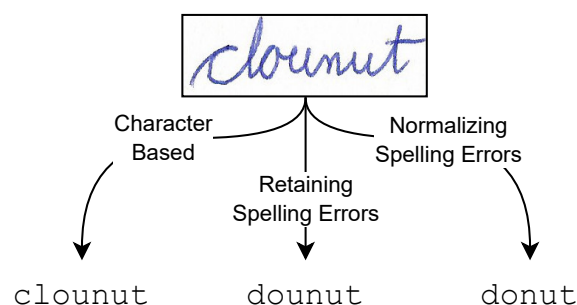


Figure 1: Handwritten example for different hypotheses (e.g. with and without normalizing spelling errors).

In this paper, we tackle this problem by first creating a dataset of handwritten text with transcripts retaining orthographic errors. For that purpose, we created comprehensive transcription guidelines (Gold et al., 2023) that precisely define our transcription goal. This is necessary as handwritten text contains other artifacts beyond orthographic errors, such as strikethroughs or inserts that we need to transcribe. In total, we transcribe 1,350 handwritten pages from German learners and thus create a dataset that is comparable in size to widely used English datasets like IAM (Marti and Bunke,



2002) and CVL (Kleber et al., 2013).

Given this dataset, we are then able to quantify to what extent existing baseline systems are unable to transcribe handwritten text, especially if we only use the underlying character recognition probabilities. We compare this with training the HWR system on parts of our data, enabling it (in theory) to learn to correctly transcribe words with orthographic errors.

Furthermore, we change the dictionary used in the HWR system to also include systematic learner errors created by an automated generator. Note that providing the actual feedback is outside the scope of this paper. Here, we focus on analyzing the problem of turning an image of handwritten text into a digitized transcript, which is currently the main obstacle to applying existing feedback methods on a scale.

## 2 Existing Datasets

For training and evaluating a handwriting recognition system that retains orthographic errors, we need a dataset combining images of learner handwriting with transcripts containing orthographic errors. To our knowledge, no such dataset exists.

IAM and CVL are mostly in English and are often used to evaluate handwriting recognition systems. IAM in its version 3.0 is an extensive dataset and consists of about 1,500 pages with more than 13,000 text lines written by 650 adults, with different segmentation levels and corresponding transcripts. CVL is comparable to IAM with about 1,600 pages from 310 adult writers. The set consists of six English and one German text and thus has a slightly increased alphabet as the German Umlauts (ä, ö, and ü) are included. In comparison to IAM, it is only transcribed word-wise, ignoring most punctuation marks or strikethrough words, although a segmentation of text lines is available.

The Growth-In-Grammar GIG dataset (Durrant and Brenchley, 2018) is a learner dataset that retained orthographic errors. However, the corresponding image data is not available.

In contrast to GIG, FD-LEX (Becker-Mrotzek and Grabowski, 2018) is another learner dataset with published image data. In comparison to IAM and CVL where the participants copied a presented text by hand, this dataset consists of texts that were freely written based on a picture or a short story, and thus, more errors were made. Albeit, the transcripts from the FD-LEX dataset normal-

Set	GYM_5	GYM_9	IGS_5	IGS_9	Sum
1	144	90	84	72	390
2	102	96	84	108	390
3	132	138	114	60	444
4	120	138	90	90	438
5	156	132	72	84	444
6	162	120	96	114	492
7	168	144	132	120	564
8	150	132	120	120	522
9	138	144	126	114	522
10	138	144	132	132	546
11	150	120	108	90	468
12	144	84	108	72	408
<b>Test Set</b>	91		Total:		5628
<b>Annotator 1</b>	168				
<b>Annotator 2</b>	1092				

Table 1: Statistics of the complete FD-LEX Dataset and our transcription effort. Cells in green are subsets for the test set; dark orange and blue are transcribed by Annotator 1 and Annotator 2, respectively.

ize orthographic errors and ignore other noise (e.g. strikethroughs).

In conclusion, none of the existing datasets fulfills our need for available image data and a transcript containing orthographic errors.

## 3 Dataset Creation

As no suitable dataset is available, we need to build one. We decided to use the German learner corpus FD-LEX as a starting point, as it already contains scans of learner handwriting with a sufficient number of orthographic errors. Looking at the example in Figure 2, we can see additional typical challenges for automatic handwriting recognition e.g. strikethroughs and inserts.

FD-LEX was built as a corpus for analyzing the writing competence of learners. It covers two different German school types: *Gymnasium* (GYM) (‘academic track school’) and *Integrierte Gesamtschule* (IGS) (‘comprehensive school’) from two grades (5<sup>th</sup> and 9<sup>th</sup>) each. It has about 5,600 scanned color pages from about 940 children and is thus exceeding the IAM (1,500 pages) and CVL (1,600 pages) datasets in size. A detailed listing can be seen in Table 1. As stated, the transcript provided with the corpus was created under another focus (e.g. normalizing orthographic errors), thus we had to transcribe it anew.

### 3.1 Transcription Guidelines

We first created transcription guidelines (Gold et al., 2023) to formulate rules on how to deal with different situations while creating an authentic transcrip-

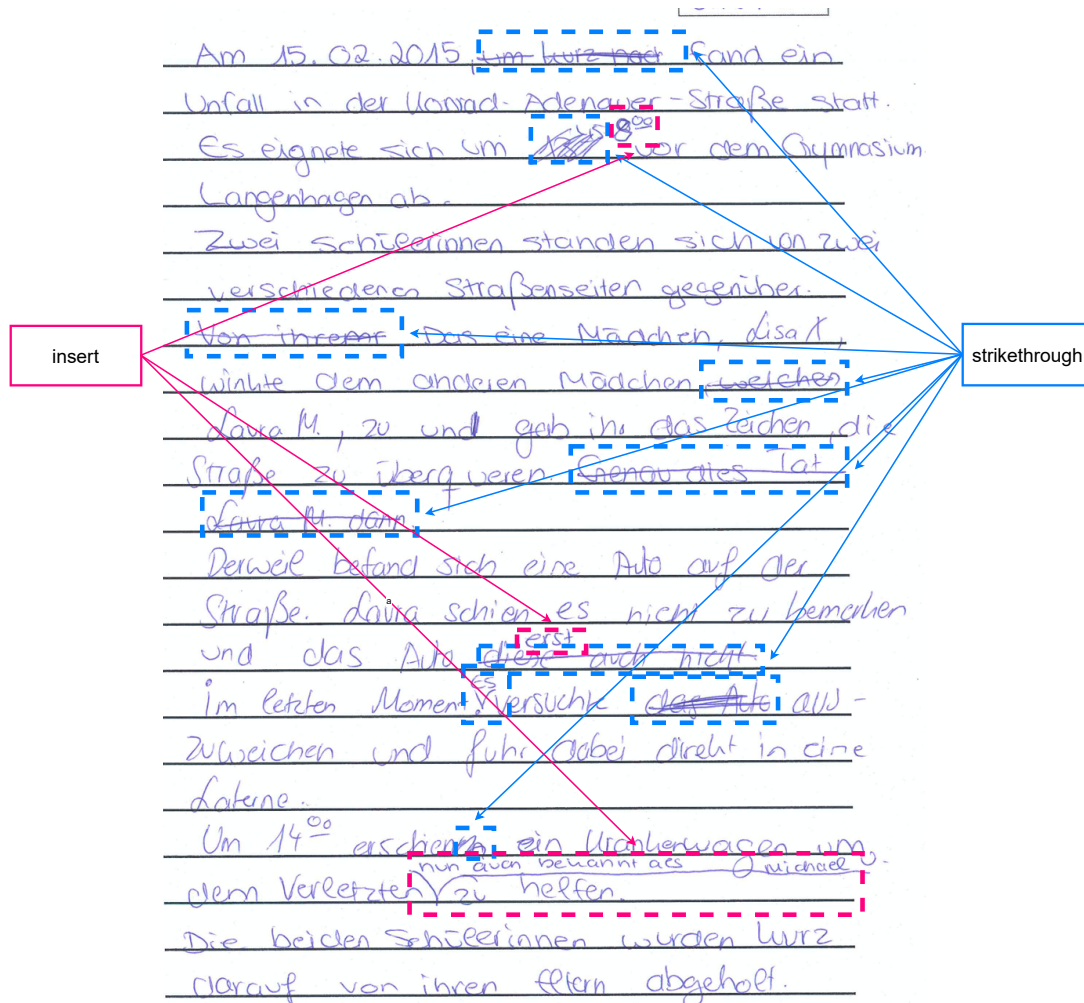


Figure 2: An example of FD-LEX with common transcription challenges like strikethroughs and inserts.

tion of the written form.<sup>1</sup> Following the guidelines should yield an exact transcript of the handwritten forms while at the same time allowing conversion into readable text automatically. This approach ensures that the transcribed text accurately reflects the writing skills of the learner and enables researchers to identify any patterns or issues related to spelling deficiencies.

We now describe the main issues covered in the guidelines:

**Text/line alignment** One line of text in the image must correspond to the line of text in the transcript.

**Content** Only the handwritten content of the learner should be transcribed. This excludes the printed text of the paper sheet as well as drawn figures.

<sup>1</sup>The transcription guidelines can be found at <https://github.com/catalpa-cl/learner-handwriting-recognition>.

**Indistinct characters** must be placed within curly brackets {}. When in doubt between two characters, the transcription should reflect the character that is appropriate in the given context. Learners may attempt to deceive teachers when uncertain whether a word should begin with a capital letter<sup>2</sup> or not, resulting in both versions being written on top of each other. In such cases, both letters should be enclosed in curly brackets and separated by a plus (+) sign, with the first letter in curly brackets being the correct one in the context.

**Spacing** should be carefully analyzed and considered in the context of the individual writing style. In cases where a gap between characters of the same word is noticeably larger than the average space between words, the spacing should be transcribed within curly brackets to indicate the deviation from the norm: {S }chool.

<sup>2</sup>Particularly, since nouns are capitalized in German.

**Spelling errors** are transcribed exactly as they appear in the original text, without any correction or modification.

**Strikethrough characters, words, lines** When a character or a word is struck through, the transcript should represent the number of characters with a hash sign (#). If a line is made invalid in the same manner, the line is transcribed with three hash signs (###).

**Inserts** Direct inserts should be transcribed enclosed in curly brackets with a less-than sign, like `< text`. Indirect inserts, which are written at a different location such as at the end of a page, can be indicated by an asterisk (\*) and a number if there are multiple inserts. These indirect inserts should be transcribed where they appear in the image. To do this, an `{insert1 *}` tag is added in the line where the text should be inserted, and the actual insert content is transcribed at the location where it appears with: `{insert1 text}`.

**Punctuation marks, special characters, emoticons** All punctuation marks have to be transcribed as they appear, with the only exception that they should align with grammar rules in regard to spacing: correct: (However,) incorrect: (However\_ ,). Special characters are treated individually for e.g. tally marks<sup>3</sup> are transcribed with an ampersand (&) `{ | & }`.

While using special signs and encoding (e.g. at inserts or tally marks, strikethroughs), a conversion between different target transcriptions can be achieved, e.g. a) for a line-wise transcript of the genuine content to be used for HWR; or b) for a coherent text where inserts are inserted and the text-line alignment is broken up to be used for semantic analysis.

### 3.2 Annotation Process

Following the guidelines, we re-transcribed about 1,250 pages, each by one annotator. To diversify our dataset, we transcribed the first 3 sets of each school type and grade (colored cells of Table 1). To assess the quality of the transcripts, some pages were transcribed by both annotators and the inter-annotator agreement (IAA) was computed. The double-annotation was done repeatedly during the whole transcription period and differences between the transcripts were discussed among annotators.

<sup>3</sup>To keep track of word counts, the learners use vertical strokes after every ten words. We refer to them as tally marks.

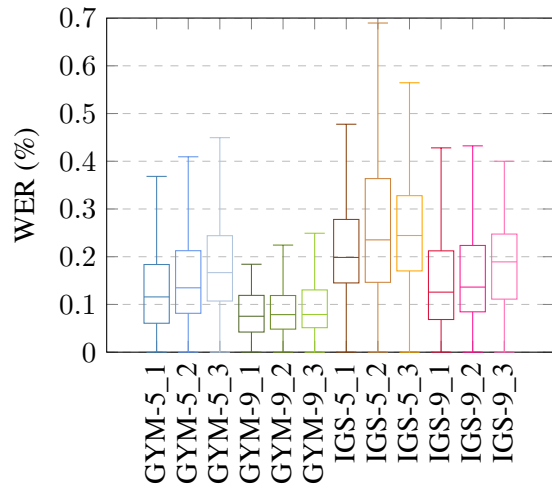


Figure 3: Distribution of Word Error Rates (WER) between the original FD-LEX dataset transcription and our error-retraining transcription.

In this way, a total of about 90 pages (subparts in green, see Table 1) were transcribed in parallel and both transcripts were merged into a gold transcription by an adjudicator. We achieved an IAA between both annotators of .98 on the character level and an IAA of .99 between both annotators and the gold label.<sup>4</sup>

### 3.3 Dataset Analysis

Transcribing the data allowed us to examine the distribution of orthographic errors, i.e. spelling, word separation, and capitalization. For that purpose, we aligned our new transcripts with the original transcripts using word alignment and measured the *word error rate* (WER). As strikethroughs are words that were made invalid, they would only increase WER and thus were excluded from our analysis.

In Figure 3, it can be observed that there are many differences between our transcripts and the original transcripts, suggesting that the use of the original transcripts may not be ideal for HWR. Additionally, the results in Figure 3 show that the 9th grade had fewer errors compared to the 5th grade, while the GYM performed better than the IGS for both grades.

## 4 Baseline Experiments

To track our recognition performance improvements, we create a baseline by training a straightfor-

<sup>4</sup>While some characters may appear unclear to one annotator and the other annotator may see it differently, we decided to calculate the IAA by ignoring curly brackets.

ward handwriting recognizer on our dataset. Commonly, the performance of the recognizer is evaluated with two metrics, namely *character error rate* (CER) and *word error rate* (WER). While CER gives numerical feedback on how many characters have been misread by the recognizer, WER measures how many words are different from the gold-standard transcription. This means that lower values indicate better recognition performance. For the purpose of this paper’s focus on word-level analysis, we will concentrate on WER rather than CER.

#### 4.1 Recognizer Setup

For our experiments, we use a recognizer based on a *convolutional neural network* (CNN) architecture combined with a *connectionist temporal classification* (CTC) (Graves et al., 2006) for decoding. The designed architecture reduces the text-line images from 2048x128 to 128x96 (Time-steps x Charset) in 7 CNN-layers, 2 BLSTMs, and a final dense layer. This architecture is based on Scheidl (2018), with CTC decoding and additional *word beam search* (WBS) for language-model decoding (Scheidl et al., 2018)<sup>5</sup>. We extended the character set used in the recognizer from 80 to 95 characters to cover all German Umlauts (‘Ä’, ‘Ö’, ‘Ü’, ‘ä’, ‘ö’, ‘ü’) and ‘ß’ as well as additional punctuation marks and special characters like ‘€’.

We use a text-line level recognizer and thus need a text-line segmentation. Thus, we first reduced the colored scans to gray level and removed ruled lines as proposed by Gold and Zesch (2022). To segment the full pages into text-lines we use a segmentation with the  $A^*$  path finding algorithm. This algorithm works on a binary image and tries to find a path through the text lines while avoiding crossing handwritten strokes.

#### 4.2 Baseline Setup

To train the recognizer we first used as much data as possible and combined IAM (~11,300 lines) and CVL (~13,400 lines) with our dataset (~12,200 lines). Furthermore, we use the gold transcripts which were transcribed by both annotators. These 91 pages (see Table 1) contain about 1,000 text-lines and are referred to as test set in the following. With the described setup and the combined training data, the recognition performance results in a CER of 11.5% and a WER of 37.6% on our test set.

<sup>5</sup><https://github.com/githubharald/SimpleHTR>, <https://github.com/githubharald/CTCWordBeamSearch>

As our dataset matches IAM and CVL in size, we decided to train the recognizer again based on our dataset only (without IAM and CVL). With this setup, we were able to improve the recognition performance slightly with a CER of 10.7% and a WER of 34.7% on our test set. With these recognition results, we decided to use this setup as our Baseline (Table 2).

### 5 Decoding with Dictionary Constraint

Most research and publicly available databases for HWR pertain to adults. In these cases, spelling errors are typically ignored because they are estimated to be rare and not important to be kept in the output. Therefore, the predicted words can be mapped to a large dictionary of possible words, which has been shown to yield better recognition rates, as recognition errors can be eliminated this way (Scheidl et al., 2018).

#### 5.1 Path Decoding and Word Beam Search

The standard method to map the Neural Network (NN) results to a text string is the CTC (Graves et al., 2006). In a more detailed manner, the NN returns a matrix containing the probability distribution for each character along so-called time-steps along the line of text. The matrix is then further analyzed by a beam search decoder such as the vanilla beam search by Hwang and Sung (2016).

However, without deeper knowledge, the beam search algorithm could randomly output an indistinguishably written character like ‘a’ as ‘o’, if the probability is the same. To avoid this, a commonly employed approach involves constraining the generated output to words that are contained in a pre-defined dictionary. This can be done with WBS as introduced by Scheidl et al. (2018).<sup>6</sup> However, with traditional dictionaries which only contain correctly spelled words, spelling errors would be eliminated from the texts.

#### 5.2 Lower Bound

The ideal dictionary would consist of the vocabulary of the learners as well as the orthographic variants. To find out what the performance would be with such an ideal dictionary, i.e. to determine the lower bound for WER that would be possible with such a dictionary, we compiled a dictionary

<sup>6</sup>Although the proposed algorithm of WBS includes a more sophisticated language model, we did not make use of it as the dictionary is increased enormously and thus increases the computational costs.

from our transcripts of the test set. This means that this dictionary only contains words that appear in the texts to be recognized as well as the specific orthographic variants that are present in the texts.

Using this dictionary in the WBS decoder, we can reduce the WER from 34.7% to 25.0%. Compared to the baseline, this is an improvement of the WER of 10 percentage points, i.e. almost one-third. With the ideal dictionary, further recognition improvements could only be achieved by changing the model or training data. This means, that the achieved performance can be seen as the Lower Bound that we want to approach.

### 5.3 German Learner Dictionary

For our purpose, we need a German dictionary covering the vocabulary of young learners in the first place. We decide to use childLex (Schroeder et al., 2015) for this purpose.<sup>7</sup> The childLex corpus was created by extracting word forms from over 500 children’s books with a target age between 6 and 12 years. Although this age range does not cover the 9th-grade students from our dataset, it seems better suitable than a dictionary compiled from adult language. To slightly restrict the extensive vocabulary, we use a subset that comprises all word forms that occurred in at least ten different books (an arbitrary cutoff point)<sup>8</sup>. This is supposed to exclude rare and specialized words, which could distract the recognizer from choosing words that are generally much more likely to appear in a text. In total, the dictionary compiled this way contains about 45,000 word forms.

Using this dictionary in Word Beam Search, i.e. constraining the output possibilities to the dictionary words, resulted in a WER of 29.6%, which is an improvement of 5 percentage points compared to the baseline, see Table 2, row ‘WBS childLex’.

### 5.4 Specific Dictionary

Since childLex is a generic dictionary compiled from books, it does not cover the whole vocabulary of the FD-LEX dataset. Therefore, we compiled another dictionary from the original transcripts of the FD-LEX dataset (in which orthographic errors were normalized) with a total of ~11,850 words. Although the dictionary is smaller than the one compiled from childLex, it benefits from contain-

<sup>7</sup>For the English community we want to mention a similar corpus <https://www.sketchengine.eu/oxford-childrens-corpus/>.

<sup>8</sup>More precisely, if a word form is included, all related word forms with the same lemma are included as well.

	CER	WER
Baseline	10.7	34.7
WBS childLex	11.3	29.6
WBS childLex + SP	10.0	30.1
WBS FD-LEX	12.4	31.3
WBS FD-LEX + SP	9.9	29.0
WBS childLex + FD-LEX + SP	9.0	25.9
Lower Bound	10.8	25.0

Table 2: Results obtained with and without using the WBS, and using different dictionaries. SP indicates dictionaries that are expanded to include spelling errors.

ing only words which the learners wrote in relation to the topics of the dataset. For example, one of the texts is about an accident with a cyclist and therefore, 20 compound words containing the German word for ‘bicycle’ appear in the dictionary, whereas only 9 such words appear in the childLex dictionary. Overall, there is an overlap of about 7,150 words between the FD-LEX dictionary and the childLex dictionary.

Incorporating the FD-LEX dictionary instead yielded a notable improvement in recognition performance at the word level compared to the baseline, achieving a WER of 31.3%, see Table 2, row ‘WBS FD-LEX’. However, it fell slightly short of the recognition accuracy obtained with the childLex dictionary.

## 6 Spelling Error Generator

To approximate the Lower Bound (see Section 5.2), spelling variants must be added to the dictionary. Thus, we generate possible (systematic) spelling errors based on the procedure described in Laarmann-Quante (2016). We generate possible misspellings for all words in the childLex and FD-LEX dictionaries. The error generation procedure works as follows: A correctly spelled word is automatically enriched with linguistic information such as phonemes, syllables, and morphemes, based on the web service G2P of the Bavarian Archive of Speech Signals (BAS) (Reichel, 2012; Reichel and Kisler, 2014)<sup>9</sup>, see also Laarmann-Quante et al. (2019a) for more information about these annotations. The information is then used to analyze (via a set of rules) which systematic errors could be made on this word. By systematic we mean that particular

<sup>9</sup><https://clarin.phonetik.uni-muenchen.de/BASWebServices/interface/Grapheme2Phoneme/>

principles of German orthography are violated, e.g. consonant doubling (\**komen* for *kommen*, eng.: ‘to come’)<sup>10</sup>, a syllabic principle, or final devoicing (\**Walt* for *Wald*, eng.: ‘forest’), a morphological principle (see Eisenberg, 2006 for the theoretical framework). We also generate errors reflecting the overuse of such principles, e.g. \**Walld* for *Wald*. Errors that cannot be explained via such principles (such as a seemingly random omission of a letter as in \**Wad* for *Wald*) are not generated because there is an infinite number of ways in which a word could be misspelled. We assume, however, that using the systematic errors in the sense described above, should capture most of the errors that the pupils commit because they are the major obstacles when learning how to spell in German.

In total, 57 different error categories can be generated (not all apply to each word, though, while some words may contain multiple instances of the same error category, e.g. when there are two doubled consonants in one word such as *Wasserfall*, eng.: ‘waterfall’). The error categories that can be generated can be found in Laarmann-Quante et al. (2019b).<sup>11</sup>

Of course, more than one error can be committed within a word. We account for this by including all possible combinations of up to 2 systematic errors that apply to a word. Including *all* possible error combinations would lead to an exponential increase of misspellings to consider, most of which will be highly unlikely, though.

## 6.1 Coverage of the Dictionaries

Applying the spelling error generation to all words in a dictionary results in an enormous increase in the number of word forms. As shown in Table 3, for the *childLex* dictionary, the number of words rises from 45,000 (row 2) to about 14 million (row 3). Likewise, *FD-LEX* with 11,000 words (row 4) rises to 3.6 million words (row 5).

As we see in the last column of the table, the original dictionaries only cover 74% (*childLex*) or 88% (*FD-LEX*) of the word forms present in the test set. Including the generated spelling errors, the coverage increases by 7-8 percentage points. However, even if *FD-LEX* and *childLex* and the spelling errors are combined (row 6 in Table 3), not all word forms are covered (90%).

<sup>10</sup>We mark misspellings with an asterisk (\*) in this paper.

<sup>11</sup>Under the levels PGI and PGII (‘Phoneme-Grapheme Correspondence Level’), SL (‘Syllabic Level’), and MO (‘Morphemic Level’)

Dictionary	# Words	Coverage
test set	1,472	100
<i>childLex</i>	45,347	74
<i>childLex</i> + SP	13,993,376	82
<i>FD-LEX</i>	11,874	81
<i>FD-LEX</i> + SP	3,670,962	88
<i>FD-LEX</i> + <i>childLex</i> + SP	15,990,735	90
<i>FD-LEX</i> + <i>childLex</i> + SP + Case	-	94

Table 3: Number of words and coverage of the test set vocabulary (in percent) for various dictionary settings. *SP* indicates dictionaries with added spelling errors, *Case* indicates that letter case variants are considered.

A manual inspection showed that one reason that not all vocabulary was covered, is that words may be capitalized at sentence beginnings in the texts, but the dictionaries do not contain capitalized variants of all words. However, including upper- and lowercase variants for all words would nearly double the size of the vocabulary, which is computationally not feasible for WBS. However, it shall be mentioned that the inclusion of both letter cases increases the coverage rate to approximately 94% (row 7 in Table 3).

We further investigated the last 6% of missing coverage, which is 88 words. 30 of these were caused by incorrect word separation (14 words that were incorrectly written together; 9 interrupted words due to line-breaks; 5 separated words due to strict transcription (e.g. huge gap after the first character); and 2 miscellaneous cases). Another 24 words were not covered due to a missing letter and 3 times two letters were swapped. These are ‘unsystematic’ errors that were not generated. For 19 words, the errors were not covered by the generator but they appeared systematic in a sense that one may think of further rules to generate them in the future, e.g. if ‘i’ follows ‘l’ the learner tends to write ‘di’ instead of ‘li’. The few words left were not covered for various reasons, e.g. interference with transcription rules, more than 2 errors in the word, and 2 non-words (number plate of a car).

## 6.2 Influence of the Advanced Dictionaries

In the following, we include the dictionaries (with and without generated spelling errors) in the decoding process of the HWR system with WBS to see if the recognition performance can be improved.

The results are shown in Table 2. We see in rows ‘WBS *childLex*’ and ‘WBS *FD-LEX*’ that including a dictionary (without spelling errors) already improves the recognition performance compared

to the Baseline by 3–4 percentage points in terms of WER.

However, adding spelling errors into the dictionary did not necessarily improve the performance. For childLex, the WER increases by 0.5 percentage points when spelling errors are added to the dictionary (compare rows 2 and 3). As discussed in Section 6.1, by adding spelling errors, the number of word forms included in the dictionary is increased extremely. Hence, chances are high that a wrong spelling variant or a spelling variant of another word is chosen. In contrast, the FD-LEX dictionary is more restricted to the vocabulary of the learners and thus could benefit from adding spelling variants: The recognition performance is increased by 1 percentage point when compared to the dictionary without spelling errors (see rows 4 and 5).

The best result was achieved by combining both dictionaries and their spelling errors. This way, the WER decreases to 25.9% and is thus within 1 percentage point of the Lower Bound.

## 7 Conclusion and Further Work

In this paper we tackled the issue of retaining orthographic errors when automatically recognizing learner handwriting. This is a prerequisite for giving automated feedback on spelling performance based on handwritten texts.

We created a handwriting recognition dataset of German learner texts based on the FD-LEX dataset by transcribing 1,350 pages using new transcription guidelines. The utilization of a dictionary to restrict the output resulted in an improvement of our baseline. Furthermore, our results indicate that incorporating generated spelling errors leads to an improvement in recognition performance at the word level, with the error rate decreasing from 35% to 25%, representing a decrease of 10 percentage points.

Although we were able to cover 94% of the originally used words using a spelling error generator, the huge number of words in the dictionary raises questions about its practicality. Therefore, one of the next goals should be to allow more probable errors while avoiding overwhelming the dictionary. Therefore, further analysis is necessary to determine which errors were made by learners in FD-LEX and which ones were addressed by the generated errors. This information can be used to reduce the size of the error set by eliminating unnecessary or rare errors. Additionally, an analysis of com-

mon error combinations can aid in generating more targeted errors while avoiding redundant ones.

Furthermore, the focus of this study was not on improving the recognition model itself. However, recognition improvements could be made by implementing a more sophisticated model like full page recognition as introduced by Bluche et al. (2017).

## Acknowledgments

This work was partially conducted at “CATALPA - Center of Advanced Technology for Assisted Learning and Predictive Analytics” of the Fern-Universität in Hagen, Germany.

## References

- Michael Becker-Mrotzek and Joachim Grabowski. 2018. FD-LEX (Forschungsdatenbank Lerner-texte). Textkorpus Scriptoria. Köln: Mercator-Institut für Sprachförderung und Deutsch als Zweitsprache. Available at: <https://fd-lex.uni-koeln.de>, DOI: 10.18716/FD-LEX/861.
- Kay Berkling and Rémi Lavalley. 2015. WISE: A Web-Interface for Spelling Error Recognition for German: A Description and Evaluation of the Underlying Algorithm. In *GSCL*, pages 87–96.
- Théodore Bluche, Jérôme Louradour, and Ronaldo Messina. 2017. Scan, Attend and Read: End-to-end Handwritten Paragraph Recognition with MDLSTM Attention. In *14th International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 1050–1055. IEEE.
- Jérémy Danna, Marieke Longcamp, Ladislav Nalborczyk, Jean-Luc Velay, Claire Commengé, and Marianne Jover. 2022. Interaction between Orthographic and Graphomotor Constraints in Learning to Write. *Learning and Instruction*, 80:101622.
- P. Durrant and M. Brenchley. 2018. *Growth in Grammar Corpus*.
- Peter Eisenberg. 2006. *Das Wort*, 3rd edition, volume 1 of *Grundriss der deutschen Grammatik*. J.B. Metzler, Stuttgart.
- Christian Gold, Ronja Laarmann-Quante, and Torsten Zesch. 2023. Preserving the Authenticity of Handwritten Learner Language: Annotation Guidelines for Creating Transcripts Retaining Orthographic Features. In *1st Computation and Written Language (CAWL) Workshop at ACL*.
- Christian Gold and Torsten Zesch. 2022. CNN-Based Ruled Line Removal in Handwritten Documents. In *18th International Conference on Frontiers of Handwriting Recognition (ICFHR)*, pages 530–544. Springer.

- Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.
- Kyuyeon Hwang and Wonyong Sung. 2016. Character-Level Incremental Speech Recognition with Recurrent Neural Networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5335–5339. IEEE.
- Firat Kizilirmak and Berrin Yanikoglu. 2022. CNN-BiLSTM model for English Handwriting Recognition: Comprehensive Evaluation on the IAM Dataset.
- Florian Kleber, Stefan Fiel, Markus Diem, and Robert Sablatnig. 2013. CVL-Database: An Off-line Database for Writer Retrieval, Writer Identification and Word Spotting. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 560–564. IEEE.
- Ronja Laarmann-Quante. 2016. [Automating Multi-Level Annotations of Orthographic Properties of German Words and Children’s Spelling errors](#). In *Proceedings of the 2nd Language Teaching, Learning and Technology Workshop (LTLT)*, pages 14–22.
- Ronja Laarmann-Quante. 2017. Towards a Tool for Automatic Spelling Error Analysis and Feedback Generation for Freely Written German Texts Produced by Primary School Children. In *7th International Workshop on Speech and Language Technology in Education (SLaTE)*, pages 36–41.
- Ronja Laarmann-Quante, Stefanie Dipper, and Eva Belke. 2019a. [The Making of the Litkey Corpus, a Richly Annotated Longitudinal Corpus of German Texts Written by Primary School Children](#). In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 43–55, Florence, Italy. Association for Computational Linguistics.
- Ronja Laarmann-Quante, Anna Ehlert, Katrin Ortman, Doreen Scholz, Carina Betken, Lukas Knichel, Simon Masloch, and Stefanie Dipper. 2019b. [The Litkey Spelling Error Annotation Scheme: Guidelines for the Annotation of Orthographic Errors in German Texts](#). *Bochumer Linguistische Arbeitsberichte (BLA)*.
- Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. 2021. TrOCR: Transformer-Based Optical Character Recognition with Pre-Trained Models. *arXiv preprint arXiv:2109.10282*.
- U-V Marti and Horst Bunke. 2002. The IAM-Database: An English Sentence Database for Offline Handwriting Recognition. *International Journal on Document Analysis and Recognition (IJ DAR)*, 5(1):39–46.
- Kathryn P Mathwin, Christine Chapparo, and Joanne Hinnit. 2022. Children with Handwriting Difficulties: Developing Orthographic Knowledge of Alphabet-Letters to Improve Capacity to Write Alphabet Symbols. *Reading and Writing*, 35(4):919–942.
- Arthur Flor de Sousa Neto, Byron Leite Dantas Bezerra, and Alejandro Héctor Toselli. 2020. Towards the Natural Language Processing as Spelling Correction for Offline Handwritten Text Recognition Systems. *Applied Sciences*, 10(21):7711.
- Karen Ray, Kerry Dally, Leah Rowlandson, Kit Iong Tam, and Alison E Lane. 2022. The Relationship of Handwriting Ability and Literacy in Kindergarten: A Systematic Review. *Reading and Writing*, pages 1–37.
- Uwe D. Reichel. 2012. [PermA and Balloon: Tools for String Alignment and Text Processing](#). In *INTER-SPEECH*.
- Uwe D. Reichel and Thomas Kisler. 2014. [Language-Independent Grapheme-Phoneme Conversion and Word Stress Assignment as a Web Service](#). In R. Hoffmann, editor, *Elektronische Sprachverarbeitung: Studententexte zur Sprachkommunikation 71*, pages 42–49. TUDpress.
- Harald Scheidl. 2018. [Build a Handwritten Text Recognition System Using TensorFlow - A Minimalistic Neural Network Implementation which can be Trained on the CPU](#).
- Harald Scheidl, Stefan Fiel, and Robert Sablatnig. 2018. Word Beam Search: A Connectionist Temporal Classification Decoding Algorithm. In *International Conference on Frontiers of Handwriting Recognition (ICFHR)*, pages 253–258. IEEE.
- Sascha Schroeder, Kay-Michael Würzner, Julian Heister, Alexander Geyken, and Reinhold Kliegl. 2015. childLex: A Lexical Database of German Read by Children. *Behavior Research Methods*, 47:1085–1094.
- Shanyu Xiao, Liangrui Peng, Ruijie Yan, and Shengjin Wang. 2020. Deep Network with Pixel-Level Rectification and Robust Training for Handwriting Recognition. *SN Computer Science*, 1:1–13.



# ExASAG: Explainable Framework for Automatic Short Answer Grading

**Maximilian Törnqvist**

Dept. of Computer and Systems Sciences,  
Stockholm University

maximilian.tornqvist@dsv.su.se

**Mosleh Mahamud**

Dept. of Computer and Systems Sciences,  
Stockholm University

mosleh.mahamud@dsv.su.se

**Erick Mendez Guzman**

Dept. of Computer Science,  
Manchester University

erick.mendezguzman@manchester.ac.uk

**Alexandra Farazouli**

Dept. of Education,  
Stockholm University

alexandra.farazouli@edu.su.se

## Abstract

As in other NLP tasks, Automatic Short Answer Grading (ASAG) systems have evolved from using rule-based and interpretable machine learning models to utilizing deep learning architectures to boost accuracy. Since proper feedback is critical to student assessment, explainability will be crucial for deploying ASAG in real-world applications. This paper proposes a framework to generate explainable outcomes for assessing question-answer pairs of a Data Mining course in a binary manner. Our framework utilizes a fine-tuned Transformer-based classifier and an explainability module using SHAP or Integrated Gradients to generate language explanations for each prediction. We assess the outcome of our framework by calculating accuracy-based metrics for classification performance. Furthermore, we evaluate the quality of the explanations by measuring their agreement with human-annotated justifications using Intersection-Over-Union at a token level to derive a plausibility score. Despite the relatively limited sample, results show that our framework derives explanations that are, to some degree, aligned with domain-expert judgment. Furthermore, both explainability methods perform similarly in their agreement with human-annotated explanations. A natural progression of our work is to analyze the use of our explainable ASAG framework on a larger sample to determine the feasibility of implementing a pilot study in a real-world setting.

## 1 Introduction

Assessment is fundamental to any educational process as an evaluation system reflecting individual performance and a way to compare results across populations (Harlen et al., 1992). Two key elements to consider when designing an assessment are question type and grading method (Gardner, 2012). While questions may come in various forms,

such as multiple-choice questions, short answers, or essays, the grading method can be either manual grading performed by domain experts or automatic grading by computational methods (Broadfoot and Black, 2004).

Previous research has established that assessing free-text short answers is a process that, besides being time-consuming, may lead to inequalities due to the difficulties in applying consistent evaluation criteria across answers (Page, 1994; Gardner, 2012). Data from several studies suggest that teachers dedicate approximately 25% to 30% of their time grading written examinations (Broadfoot and Black, 2004; Sukkarieh et al., 2003). Moreover, manual grading requires concentration for long periods of time, which could lead to differences in grading for answers with similar quality, creating inequities in the assessment process and its outcome (Whittington and Hunt, 1999; Burrows et al., 2015).

In the literature, automatic short answer grading (ASAG) is defined as the task of assessing short natural language responses to objective questions using computational methods (Page, 1994; Whittington and Hunt, 1999). ASAG techniques have evolved from traditional rule-based models to state-of-the-art systems utilizing deep learning-based natural language processing (NLP) models (Sukkarieh et al., 2003; Leacock and Chodorow, 2003; Galhardi and Brancher, 2018). Researchers have been able to build supervised learning models based on assessment questions, answers provided by students, and the corresponding grades assigned by teachers (Burrows et al., 2015; Willis, 2015). The objective is, therefore, to predict which label score a new question-answer pair should achieve.

Over the past five years, researchers have leveraged the power of novel deep learning architectures such as the Transformer (Vaswani et al., 2017) to

improve accuracy for ASAG models (Sung et al., 2019a). Nevertheless, the performance improvement has come at the cost of models becoming less understandable for stakeholders, and their opaqueness has become an obstacle to their deployment in the educational domain (Belle and Papantonis, 2020; Arrieta et al., 2020). Consequently, Explainable Artificial Intelligence (XAI) has emerged as a relevant research field aiming to develop methods that allow stakeholders to understand the outcome of deep learning-based systems (Gunning et al., 2019; Arrieta et al., 2020). As such, several lines of evidence suggest that providing insights into models' inner workings might be helpful in building trust in these systems and detecting potential biases (Belle and Papantonis, 2020; Arrieta et al., 2020; Jacovi and Goldberg, 2021).

A great deal of previous research into XAI methods for explaining NLP models has focused on building reliable associations between the input text and output label and quantifying how much each element (e.g., word or token) contributes to the final prediction (Danilevsky et al., 2020). Such XAI methods can usually be divided into feature importance-based explanations (Simonyan et al., 2013), perturbation-based explanations (Zeiler and Fergus, 2014), explanations by simplification (Ribeiro et al., 2016) and language explanations (Lei et al., 2016). Previous studies have indicated that *rationales* or language explanations are easier to understand and use since they are verbalized in human-comprehensible natural language (Lei et al., 2016; DeYoung et al., 2019).

This study focuses on explaining binary text classification for student responses gathered from a Data Mining course exam. As such, the main objective is to generate a framework that predicts binary grades and simultaneously produces associated rationales in order to justify the predicted grade of a given student response. By doing so, we intend to enrich the insights given by previous research, by presenting a framework that demonstrates how recent progressions of deep learning architectures and XAI can be combined in order to address the problem of ASAG. As such, we aim to set an example for how future research can incorporate XAI in the educational domain. Conclusively, our main contributions are as follows:

1. Suggesting a framework for creating sentence-level and word-level attributions by utilizing token-level relevancy scores.

2. Evaluating contemporary explainability methods by measuring the Intersection-Over-Union of our language explanations and human rationales.
3. Applying a fine-tuned Transformer model to perform ASAG on data-scientific question-answer pairs by utilizing collected data from a course in Data Mining.

## 2 Related Work

Large Language Models (LLMs) such as Transformer models have been increasingly applied in the domain of ASAG (Haller et al., 2022). Given a limited amount of examples, Transformer models such as BERT have proven their capability to achieve state-of-the-art performance within the field of ASAG (Sung et al., 2019b). The ability to handle single short documents, such as question-answer pairs, makes BERT a suitable model for various downstream tasks (Devlin et al., 2018). Most previous research and implementations focus on the model's effectiveness using standard classification metrics such as F1 and accuracy, precision, and recall (Haller et al., 2022). However, there is a limited amount of research addressing *why* certain predictions are being made. As a consequence, a lack of trust and understanding of the model predictions remains an issue. Thus, our work explores the use of explainability techniques as a tool for ASAG, in order to increase the understanding of the predictions being done.

Rationale extraction refers to a post-hoc explainability method for NLP models in which the goal is to create deep learning-based NLP solutions explainable by uncovering part of an input sequence that the prediction relies on the most (Lei et al., 2016; DeYoung et al., 2019). Most previous research on rationale extraction has been carried out using an *encoder-decoder* architecture. In such a setting, the *encoder* works as a tagging model, where each word in the input sequence receives a binary tag indicating whether it is included in the rationale. The *decoder* then only accepts the input highlighted as a rationale and maps it to the target labels (Zaidan et al., 2007; Bao et al., 2018; Narang et al., 2020).

Previous studies have proposed a multi-task learning approach for rationale extraction utilizing two models and training them jointly to minimize a composite cost function (Lei et al., 2016; Bastings et al., 2019; Paranjape et al., 2020). Unfortunately,

one of the main drawbacks of multi-task learning architectures for rationale extraction is that it is challenging to train the encoder and decoder jointly under instance-level supervision (Zhang et al., 2016; Jiang et al., 2018). Pipelined models are a simplified version of the encoder-decoder architecture in which the encoder is first trained to extract the rationales. Then the decoder is fit to perform prediction using only the rationale (Zhang et al., 2016; Jain et al., 2020). It is important to note that no parameters are shared between the two models and that rationales extracted based on this approach have been learned in an unsupervised manner since the encoder is deterministic by nature.

There is little consensus on what makes a good machine-generated rationale and how to evaluate a rationale for benchmarking. Most researchers investigating rationale evaluation have utilized *proxy-based* methods, where rationales are assessed based on automatic metrics that attempt to measure desirable properties (Carton et al., 2020). One of the most common methods for evaluating rationales is to measure how well they agree with explanations provided by human annotators (DeYoung et al., 2019). In the context of explainable NLP, this property is referred to as *plausibility*. As such, it is usually evaluated based on the token overlap between human annotations and machine-generated rationales. Using such an approach, researchers have been able to derive token-level precision, recall, and F1 scores using Intersection-over-Union (IOU) at token level (Paranjape et al., 2020; Chan et al., 2021; Guerreiro and Martins, 2021).

### 3 Explainable Autograding Framework

The explainable framework is illustrated in Figure 1, consisting of an encoder responsible for generating explanations and a decoder responsible of performing the binary classification.

#### 3.1 Encoder

The encoder is built using two main components, where the first component corresponds to the explainability method of use, and the second component corresponds to the ranking and processing of the given attributions created by the used explainability method. The two mentioned components result in a ranking for each sentence in a student’s answer based on its importance in end classification. Thus, the concept of the framework itself is not dependent on the individual explain-

ability methods presented in this study. As such, with minor adjustments according to the outputs of the used method, the concept of the presented framework should be considered generalizable and possible to implement in conjunction with other token-based methods of attribution. Subsequently, the following paragraphs will introduce the explainability methods being used in this study.

#### 3.2 Explainability methods

As the complexity of a model increases, the model itself cannot longer be used as a method for explanation. As such SHAP utilizes cooperative game theory and Shapley values to explain a model’s output prediction (Lundberg and Lee, 2017). By doing so, SHAP creates an interpretable approximation of the original model, which is referred to as the explanation model. In essence, SHAP is a model-agnostic explainability method that captures the importance value of an input feature by perturbing the input feature and observing the change in the model’s prediction output. By observing the resulting output of the perturbation, SHAP makes it possible to assign each input feature an importance value. In practice, SHAP utilizes additive feature attributions, which in essence can be defined as a mapping of the original input features to simplified features. As such, it achieves the aforementioned interpretable approximation of the original model. In the task of ASAG, the tokens included in the answer correspond to the input features. Consequently, each and every token in the answer will be given a relevancy value.

Similarly, each input feature is assigned an attribution value with Integrated Gradients (IG). IG is an explainability method based on two main axioms; Sensitivity and Implementation Invariance (Sundararajan et al., 2017). IG measures the attribution value by comparing the model’s output function of the input with the model’s output function of an uninformative baseline. The uninformative baseline could correspond to a black image in an object recognition task, while for text classification, the baseline could correspond to a zero embedding vector. The integrated gradients can then be defined as ‘the path integral of the gradients along the straight line path from the baseline to the input’ (Sundararajan et al., 2017). For a text classification task, the integrated gradients are calculated by interpolating between the baseline and the original output for  $k$  number of steps. This

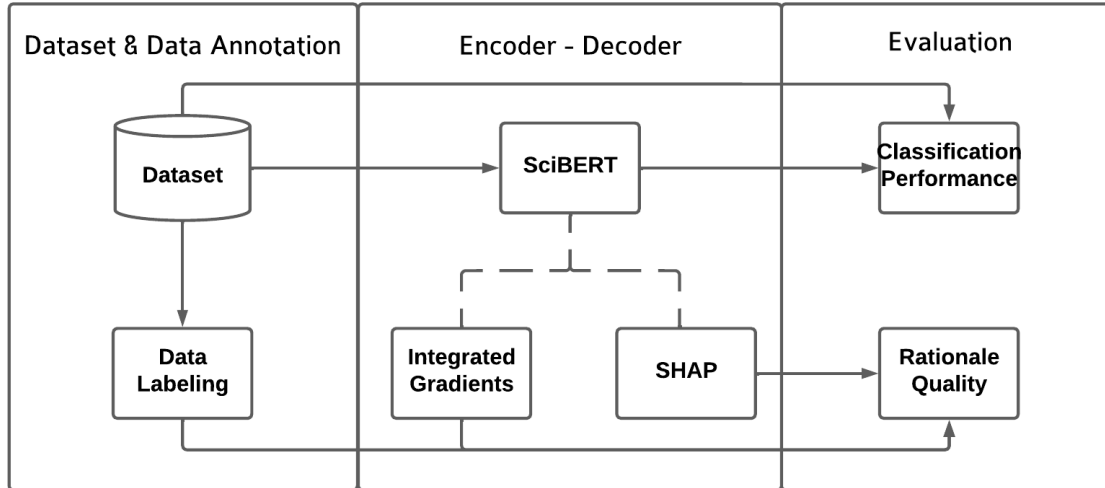


Figure 1: Explainable Framework for Automatic Short Answer Grading.

gives insight into each feature independent of the others and its impact on the output prediction. Furthermore, Gradient-based explanations are known to be robust and efficient (Nielsen et al., 2022).

The reason why SHAP and IG are used in this experiment is that SHAP can handle interactions between features when explaining (Nielsen et al., 2022). In contrast, IG only considers each feature’s individual contribution, making it suitable to observe both effects. Both post-hoc explainability methods have a reputation for being robust, and neither has an effect on the end classification accuracy (Vale et al., 2022; Lakkaraju et al., 2020) hence, making it reasonable to apply to ASAG tasks.

### 3.2.1 Sentence Level Explanations

Since Transformer models usually represent singular words as multiple tokens, explainability methods such as SHAP and IG will return attributions at a token level when used in combination with Transformers. In this framework, attributions are grouped per sentence, creating Sentence Level Attributions (SLAs). The SLAs are all based on Word Level Attributions (WLAs), which in turn are based on the original Token Level Attributions (TLAs) generated by the explainability methods.

We define the WLAs as the sum of all the TLAs representing a single word. Furthermore, as including stopwords in the SLA could lead to very neutral attribution values of sentences with a considerable amount of stopwords, we define the SLAs as the mean attribution of all non-stopwords contained in a sentence. As such, stopwords are assumed not

to be highly determining for the end classification. Thus, they are completely ignored in the calculation of SLA. Furthermore, as a consequence of the partially arithmetic characteristics of the data set and Transformers’ inability to handle such arithmetics, any non-alphabetical characters are removed before calculating the SLAs.

### 3.3 Decoder

The components of the decoder are a Transformer model fine-tuned on a exams from a data mining course, where the characteristics of the data set are further detailed in section 4.1.

**The model** For the classification of the text, we use SciBERT (Beltagy et al., 2019), as it is a pre-trained model based on the architecture of BERT, which uses a corpus of 1.14 million papers instead of the original pre-training data found in BERT. Of these 1.14 million papers, 18% of the papers in the corpus comes from the domain of computer science. In terms of representing language, the vocabulary of the SciBERT model only overlaps the vocabulary of the BERT model by 42% (Beltagy et al., 2019). As such, this difference in vocabulary illustrates the differences between scientific text in comparison to general text. Furthermore, it also highlights the importance of choosing the appropriate model and associated vocabulary depending on the domain of the given task. Given that SciBERT can be considered to be of a somewhat computer scientific

### 3.4 Evaluation

**Classification Performance** For evaluating the classification performance of the used model, we calculate precision, recall, and F-1 score against the given test set.

**Plausibility** For calculating the quality of the model rationales, we calculate the Inter-Annotator Agreement (IAA) (Zaidan et al., 2007; Carton et al., 2020) as the Intersection-Over-Union (IOU). The use of IOU is to calculate the overlap of the model rationales and the human rationales. Before calculating this overlap, we first ensure that the model rationales and the human rationales are in a comparable format. To achieve this, both of them are processed in a standardized manner.

As the explanation models used generates attribution scores for all text in the given response, the generated attributions needs to be filtered in order to conduct a fair comparisson with human rationales (as they do not usually contain all of the text in a given response). As such, the sentences can be ranked by their respective attribution value. Following such a ranking, the top  $k$  attributions corresponding to the given grade could be picked out for comparisson with the human rationales, where  $k$  is defined by the number of rationales annotated by the human. As such, the top  $k$  sentences with the highest SLA are selected for comparison with the human rationales if the label is "Satisfactory", where  $k = \text{the number of sentences in human rationales}$ . However, if the label of the answer is "Non-satisfactory", the  $k$  sentences with the lowest SLA are selected for comparison with the human rationales.

In order to compare the sentences, both set of sentences are split into non-stopword tokens. From these sets of tokens, empty strings and non-alphabetic characters are removed. Finally, the two sets of tokens will represent the model and human rationales when calculating the IAA.

## 4 Data, Experiments and Results

### 4.1 Data set

As part of a project in automatically grading exams at Stockholm University, the data selected in this study was selected in order to partly evaluate the potential of using automatically grading systems on low-resource data. As such, the selected data set used in this experiment is an English data set consisting of 1131 question-answer pairs collected from graded exams of a Data Mining course at

Stockholm University. As such, the data has been collected from a limited amount of course iterations. In total, there are 31 unique questions, with an average of 36,5 answers per question. Given the amount of question answers pairs, the adjustments and changes that have been applied to the questions inbetween the given iterations, and the amount of answers per question, it is reasonable to deem the data set to be of a low-resource charachter. In essence, this poses a fundamental challenge for building grading systems, where the amount of examinatory data can be limited due to a multitude of factors such as limited data collection, frequent adjustments to questions or course content, or due to the course being new. As such, utilizing such a data set, will help evaluate the potential of building automatically grading systems on low-resource data.

The data set also features a lot of scenario-based questions, where the student is often asked to provide a solution for a scenario-based problem. This type of response generally involves complex reasoning about the problem and as a consequence, the answers are usually long compared to answers in data sets previously used, with an average length of 155 words per answer across the whole data set. Given this, it could be argued that the task of grading these answers could be seen as a more elaborate version of the ASAG task that a lot of previous research has been focusing on (Haller et al., 2022). Furthermore, some of the question-answer pairs involve small amounts of arithmetics. Given the amount of available data and varying class representation, the scales of grading have been converted from the original scales (0-5, 0-8, and 0-10) to binary labels (0-1). From the original scales, binary labels were derived by assessing every answer that achieved 50% or more of the original maximum grade as a satisfactory(1) answer and every answer that achieved less than 50% of the original maximum grade as a non-satisfactory answer(0). Following the conversion, there are 667 satisfactory answers and 464 non-satisfactory answers.

### 4.2 Data annotation

Before performing the annotation, we developed an annotation scheme and guidelines to facilitate labeling question-answer pairs (Krippendorff, 2004). The scheme is based on the rubric associated with each question defined by Stockholm University lecturers. As mentioned before, we focused on binary

text classification. Consequently, we asked our annotators to label each item as “Satisfactory” or “Non-Satisfactory” based on whether they would assign at least 50% or more of the total maximum grade for each question. To illustrate, an answer graded 10 points to a question worth 20 points would be satisfactory, while an answer graded 9 points to the same question would be labeled as non-satisfactory. However, since our goal is to provide richer annotations that support grading, we also asked our annotators to select phrases and sentences to justify their labeling decisions (Zaidan et al., 2007; DeYoung et al., 2019; Guzman et al., 2022). The annotation guidelines and examples of our dataset are available upon request.

Since annotations of the original dataset was not available, the annotation of the corpus was completed by two annotators aged above 25 years old with degrees in Data Mining and Computer Science from Stockholm University. Considering how domain-specific our research is and the data privacy constraints of our dataset, we decided against crowd-sourcing the annotation. During the annotation process, the annotators were encouraged to ask questions over online sessions to facilitate feedback and ensure high-quality human rationales (Nowak and Ruger, 2010). In order to avoid any bias or preconceptions being passed on from the authors to the annotators during the feedback sessions, the annotations were carried out prior to the creation of any model rationales. Furthermore, in order to avoid being directly involved in any of the examples, we highly encouraged the annotators to ask questions of a conceptual character rather than to showcase specific examples from the dataset.

To validate our annotation guidelines, we randomly selected 20 question-answer pairs and asked our annotators to label them independently using LightTag (LightTag, 2018) as the annotation platform. This preliminary validation helped the annotators to familiarize themselves with the scope of the task and to understand how to use LightTag. The trial run enabled us to obtain constructive feedback on the annotation scheme and guidelines (Zou et al., 2021).

We assessed the quality of the annotations using the F1 score as IAA metric (Zaidan et al., 2007; Carton et al., 2020). Considering the aim of our research, we computed IAA at the level of binary labels and rationales (Krippendorff, 2004). Considering the annotations of our most senior annotator

(A1) as the gold standard, we obtained a micro-averaged F1 score of 0.94 for the 20 items in the trial run.

As mentioned before, measuring exact matches between rationales is likely too strict. Similarly to what we described as one of the evaluation metrics for the encoder, we used IOU at a token level (DeYoung et al., 2019). For rationales’ IAA, the IOU is the size of the token overlap of the two human-generated explanations, divided by the size of their union (Carton et al., 2020). We counted it as a match if the IOU exceeds a user-defined threshold. Following (Zaidan et al., 2007), we utilized 0.5 as the threshold and derived a micro-averaged F1 score of 0.81 for rationales in the trial run.

Several lines of evidence suggest that reaching a high IAA for rationale labeling is still challenging, mainly because of the complexity of the annotation task itself and the subjective nature of the human rationales (Lei et al., 2016; Strout et al., 2019; Carton et al., 2020). Nevertheless, we observed a fair agreement between our annotators compared with previous work on rationales for binary text classification (Zaidan et al., 2007; DeYoung et al., 2019). Consequently, we sampled 200 items from our dataset and asked each annotator to label 100 question-answer pairs to consolidate the rationale-annotated dataset to evaluate our explainable framework.

Our annotators labeled almost two-thirds of the 200 question-answer pairs as “Satisfactory” (134 items). The human rationales for the “Satisfactory” label were, on average, 55 words-length with a standard deviation of 12 words. The rationales assigned to the “Non-Satisfactory” class were slightly shorter, with an average of 48 words and a standard deviation of 18 words.

### 4.3 Experiments

For the classification experiment, the data was split using stratification into a training set consisting of 757 examples and a test set consisting of 374 examples. Using the training and test set, the model was evaluated both with fine-tuning on the training set and without any fine-tuning. The aim of this method is to demonstrate the difference that fine-tuning can make in classification performance when the amount of data is limited (for results with no fine-tuning, see Appendix A).

Given the previously described question-answer pairs, the models were fine-tuned for 3 epochs with

a batch size of 8. For optimization, AdamW was used with a learning rate of  $2e-5$  and a weight decay of 0.01.

For evaluating the performance of the classification model, a total amount of 1131 question-answer pairs were used. From these 1131 examples, 757 examples were used for fine-tuning the model, while 374 examples were used for testing the model. The metrics for measuring the performance of the classification model were precision, recall, and F1-score on a micro-level as well as on a macro-level, for both of the labels, which can be seen in Table 1.

For evaluating the performance of the explainability framework, a sample of 5 questions was chosen for this experiment. The sampling of questions was based on factors such as label distribution, the average length of answers, the number of answers per question, and the amount of arithmetics involved in the question. Since the data set was very limited in terms of the number of answers per question, we made sure that both of the class labels were represented in each of the sampled questions. Having this in mind, we also made sure not to include questions that were relatively high in arithmetical answers. The support of the individual questions ranges from 35 answers per question to 50 answers per question, with a mean of 41 answers per question. In total, the selected data set for evaluating the sentence explainability framework consisted of 200 question-answer pairs. Thus, given the limited annotation budget of the project, the explainability framework is only evaluated on a subset of the data set used for evaluating the classification task. As such, the questions with the most lengthy answers were also rejected as a part of the evaluation process.

## 5 Results

### 5.1 Classification results

Table 1 shows the classification performance of the model used in the explainability experiments, where the classification performance is evaluated using precision, recall, and F1-score. As seen in the table, there is a difference in classification performance between the two given labels. The difference in performance could be expected as a consequence of the imbalance in the data set.

Table 2 shows F1-score and recall based on a varying threshold and the number of matches between the human rationales and the model ratio-

	Precision	Recall	F1-score
Label 0	0.74	0.67	0.70
Label 1	0.79	0.84	0.82
Macro Avg	0.77	0.76	0.76

Table 1: Overall classification performance metrics of fine-tuned SciBERT, where Label 0 = Non-satisfactory and Label 1 = Satisfactory.

nales generated by IG. Where a match is registered if the IAA calculated as the IOU between the model rationales and the human rationales exceeds the given threshold. As mentioned in section 3, the calculation is carried out using two sets of tokens representing the human and model rationales. In this scenario, the ground truth will always be a match, which means that the recall will represent the number of matches made out of all possible matches. Given a Threshold of 0.5, the results show an F1-score of 0.62 and a recall of 0.45. This means that out of all possible matches, the IAA exceeds the 0.5 threshold in 45% of all answers.

Threshold	F1	Recall
0.1	0.95	0.91
0.2	0.92	0.85
0.3	0.82	0.70
0.4	0.75	0.60
0.5	0.62	0.45

Table 2: Overall performance metrics for IG, based on a threshold and the number of matches.

Table 3 shows the F-1 score and Recall based on a varying threshold and the number of matches between the human rationales and the model rationales generated by SHAP. If the IAA calculated as the IOU exceeds the threshold of 0.5 for a given answer, we calculate it as a match. Given a Threshold of 0.5, the results show an F1-score of 0.63 and a recall of 0.46. Which is similar to the results achieved by IG. This means that out of all possible matches, the IAA exceeds the 0.5 threshold in 46% of all answers.

## 6 Discussion

When comparing the F1-score and recall of the SHAP method with the F1-score and recall of the IG method there seems to be little to no difference in their respective IAA with the human annotators. However, both of the methods seem to do well given the complexity of the data as well as the lim-

Threshold	F1	Recall
0.1	0.96	0.92
0.2	0.89	0.81
0.3	0.83	0.70
0.4	0.77	0.62
0.5	0.63	0.46

Table 3: Overall performance metrics for SHAP, based on a threshold and the number of matches.

ited amount of data that was used for fine-tuning. Given the SciBERT model and these accompanying explainability methods, it seems to be possible to generate representative explanations as well as explanations that could be valuable for a human annotator.

Given that the data set used is not only considerably smaller but also considerably more complex in terms of answer length than most data sets previously used in the task of ASAG, a slight decrease in classification performance is expected compared to previous research. Furthermore, one implication of the classification results is that Transformer models seem to require a very small amount of question-specific data in order to substantially improve its performance in classification, even when given relatively complex data. However, such solutions may not replace human expertise. Rather, using a combination of these models and the presented explainability methods, this performance can increase confidence in the given explanations and as a consequence, it could help aid and assist human experts in grading when data is very limited.

## 7 Conclusion and Future Work

NLP tools hold immense potential for scoring free-text answers from students and augmenting teachers' evaluation capabilities in a scalable manner. Transformer-based models can help identify patterns from students' responses and prioritize solutions that need further checking. However, their black-box nature becomes an obstacle when deploying these models in real-world educational applications. To bridge this knowledge gap, we introduce an explainable ASAG framework that produces competitive predictions along with human-understandable natural language explanations. Our framework leverages LLMs capabilities combined with post-hoc explainability methods that do not require training, reducing the number of question-answer pairs needed to achieve state-of-the-art re-

sults.

Furthermore, the classification performance proves that LLMs can achieve competitive ASAG performance on complex questions with a low number of answers per question when given domain-specific training, indicating a low threshold for applying domain-specific ASAG. As a consequence, the resulting performance could give a certain degree of confidence when assisting teachers with valuable explanations.

Further work needs to be done to establish whether incorporating human-generated rationales during training can boost the model's predictive performance and the quality of its generated explanations (Strout et al., 2019; Lei et al., 2016). Our future work aims to incorporate them using a multi-task learning approach and evaluate rationales beyond the plausibility dimension covered in the presented article.

Finally, we hope our framework and initial results can help promote research on explainability in ASAG systems.

## 8 Limitations

Given that the WLAs are calculated as the sum of all the TLAs representing one single word, it is possible that there could be an underlying preference for longer words in the framework. However, multiple tokens in a word could also have conflicting attributions, so it is not entirely clear how this affects the framework. Given the results of this implementation, it could be reasonable to try and calculate the WLAs as the mean of all TLAs instead.

Furthermore, it is reasonable to discuss the consequences of the preprocessing steps being carried out in the experiment. Although such preprocessing steps might increase the IAA measured between the human rationales and model rationales, it is reasonable to question what these preprocessing steps actually result in and their possible value in real-world applications. In cases where the use case is to identify and highlight certain important words, such preprocessing steps might bring a considerable amount of value. However, if the end goal is to represent the model's attention as precisely as possible, these preprocessing steps might skew the representation of the model's attention. Consequently, one could argue that there exists a trade-off between usable model explanations, which can be used as an assisting or guiding tool for the human



expert, and explanations that are fair representations of the model’s inner workings. In the case of ASAG, explanations such as the ones created by the presented framework could likely be used as an assisting tool in helping human expert graders find important words or sentences. Given such a framework, the speed of grading could likely be increased without removing the trust of having a human grader making the end decision.

Lastly, it is worth noting that the use of top k sentences should only be seen as a means of calculating IAA. However, in a real-world inference setting, the number of relevant sentences might be dependent on the task as well as the subject. In the case of assisting a human expert in grading, the number of top k sentences might be a parameter controlled by the human expert in order to showcase only the most relevant sentences marked by the model annotations, where the number of relevant sentences might be dependent on the length of the student answer as well as the complexity of the given question.

## References

- Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrién Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. 2020. [Explainable artificial intelligence \(xai\): Concepts, taxonomies, opportunities and challenges toward responsible ai](#). *Information fusion*, 58:82–115.
- Yujia Bao, Shiyu Chang, Mo Yu, and Regina Barzilay. 2018. [Deriving machine attention from human rationales](#). *arXiv preprint arXiv:1808.09367*.
- Jasmijn Bastings, Wilker Aziz, and Ivan Titov. 2019. [Interpretable neural predictions with differentiable binary variables](#). *arXiv preprint arXiv:1905.08160*.
- Vaishak Belle and Ioannis Papantonis. 2020. [Principles and practice of explainable machine learning](#). *arXiv preprint arXiv:2009.11698*.
- Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. [SciBERT: A pretrained language model for scientific text](#). 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 3615–3620, Hong Kong, China. Association for Computational Linguistics.
- Patricia Broadfoot and Paul Black. 2004. [Redefining assessment? the first ten years of assessment in education](#). *Assessment in Education: Principles, Policy & Practice*, 11(1):7–26.
- Steven Burrows, Iryna Gurevych, and Benno Stein. 2015. [The eras and trends of automatic short answer grading](#). *International Journal of Artificial Intelligence in Education*, 25(1):60–117.
- Samuel Carton, Anirudh Rathore, and Chenhao Tan. 2020. [Evaluating and characterizing human rationales](#). *arXiv preprint arXiv:2010.04736*.
- Aaron Chan, Maziar Sanjabi, Lambert Mathias, Liang Tan, Shaoliang Nie, Xiaochang Peng, Xiang Ren, and Hamed Firooz. 2021. [Unirex: A unified learning framework for language model rationale extraction](#). *arXiv preprint arXiv:2112.08802*.
- Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. [A survey of the state of explainable ai for natural language processing](#). *arXiv preprint arXiv:2010.00711*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. 2018. [Bert: Pre-training of deep bidirectional transformers for language understanding](#). *arXiv preprint arXiv:1810.04805*.
- Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. 2019. [Eraser: A benchmark to evaluate rationalized nlp models](#). *arXiv preprint arXiv:1911.03429*.
- Lucas Busatta Galhardi and Jacques Duflío Brancher. 2018. [Machine learning approach for automatic short answer grading: A systematic review](#). In *Ibero-american conference on artificial intelligence*, pages 380–391. Springer.
- John Gardner. 2012. *Assessment and learning*. Sage.
- Nuno Miguel Guerreiro and André FT Martins. 2021. [Spectra: Sparse structured text rationalization](#). *arXiv preprint arXiv:2109.04552*.
- David Gunning, Mark Stefik, Jaesik Choi, Timothy Miller, Simone Stumpf, and Guang-Zhong Yang. 2019. [Xai—explainable artificial intelligence](#). *Science Robotics*, 4(37).
- Erick Mendez Guzman, Viktor Schlegel, and Riza Batista-Navarro. 2022. [Rafola: A rationale-annotated corpus for detecting indicators of forced labour](#). *arXiv preprint arXiv:2205.02684*.
- Stefan Haller, Adina Aldea, Christin Seifert, and Nicola Strisciuglio. 2022. [Survey on automated short answer grading with deep learning: from word embeddings to transformers](#). *arXiv preprint arXiv:2204.03503*.
- Wynne Harlen, Caroline Gipps, Patricia Broadfoot, and Desmond Nuttall. 1992. [Assessment and the improvement of education](#). *The curriculum journal*, 3(3):215–230.

- Alon Jacovi and Yoav Goldberg. 2021. [Aligning faithful interpretations with their social attribution](#). *Transactions of the Association for Computational Linguistics*, 9:294–310.
- Sarthak Jain, Sarah Wiegrefe, Yuval Pinter, and Byron C Wallace. 2020. [Learning to faithfully rationalize by construction](#). *arXiv preprint arXiv:2005.00115*.
- Xin Jiang, Hai Ye, Zhunchen Luo, WenHan Chao, and Wenjia Ma. 2018. [Interpretable rationale augmented charge prediction system](#). In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 146–151.
- Klaus Krippendorff. 2004. [Measuring the Reliability of Qualitative Text Analysis Data](#). *Quality and Quantity*, 38:787–800.
- Himabindu Lakkaraju, Nino Arsov, and Osbert Bastani. 2020. [Robust and stable black box explanations](#). In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5628–5638. PMLR.
- Claudia Leacock and Martin Chodorow. 2003. [C-rater: Automated scoring of short-answer questions](#). *Computers and the Humanities*, 37(4):389–405.
- Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. [Rationalizing neural predictions](#). *arXiv preprint arXiv:1606.04155*.
- LightTag. 2018. The text annotation tool for teams. <https://www.lighttag.io/>.
- Scott M. Lundberg and Su-In Lee. 2017. [A unified approach to interpreting model predictions](#). In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, page 4768–4777, Red Hook, NY, USA. Curran Associates Inc.
- Sharan Narang, Colin Raffel, Katherine Lee, Adam Roberts, Noah Fiedel, and Karishma Malkan. 2020. [Wt5?! training text-to-text models to explain their predictions](#). *arXiv preprint arXiv:2004.14546*.
- Ian E Nielsen, Dimah Dera, Ghulam Rasool, Ravi P Ramachandran, and Nidhal Carla Bouaynaya. 2022. [Robust explainability: A tutorial on gradient-based attribution methods for deep neural networks](#). *IEEE Signal Processing Magazine*, 39(4):73–84.
- Stefanie Nowak and Stefan Ruger. 2010. [How reliable are annotations via crowdsourcing: A study about inter-annotator agreement for multi-label image annotation](#). In *Proceedings of the international conference on Multimedia information retrieval*, pages 557–566.
- Ellis Batten Page. 1994. [Computer grading of student prose, using modern concepts and software](#). *The Journal of experimental education*, 62(2):127–142.
- Bhargavi Paranjape, Mandar Joshi, John Thickstun, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. [An information bottleneck approach for controlling conciseness in rationale extraction](#). *arXiv preprint arXiv:2005.00652*.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. [Model-agnostic interpretability of machine learning](#). *arXiv preprint arXiv:1606.05386*.
- Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. [Deep inside convolutional networks: Visualising image classification models and saliency maps](#). *arXiv preprint arXiv:1312.6034*.
- Julia Strout, Ye Zhang, and Raymond J Mooney. 2019. [Do human rationales improve machine explanations?](#) *arXiv preprint arXiv:1905.13714*.
- Jana Z Sukkarieh, Stephen G Pulman, and Nicholas Raikes. 2003. [Auto-marking: using computational linguistics to score short, free text responses](#).
- Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. [Axiomatic attribution for deep networks](#). In *International conference on machine learning*, pages 3319–3328. PMLR.
- Chul Sung, Tejas Dhamecha, Swarnadeep Saha, Tengfei Ma, Vinay Reddy, and Rishi Arora. 2019a. [Pre-training bert on domain resources for short answer grading](#). 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 6071–6075.
- Chul Sung, Tejas Indulal Dhamecha, and Nirmal Mukhi. 2019b. [Improving short answer grading using transformer-based pre-training](#). In *International Conference on Artificial Intelligence in Education*, pages 469–481. Springer.
- Daniel Vale, Ali El-Sharif, and Muhammed Ali. 2022. [Explainable artificial intelligence \(xai\) post-hoc explainability methods: Risks and limitations in non-discrimination law](#). *AI and Ethics*, pages 1–12.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). *arXiv preprint arXiv:1706.03762*.
- Dave Whittington and Helen Hunt. 1999. Approaches to the computerized assessment of free text responses.
- Alistair Willis. 2015. [Using nlp to support scalable assessment of short free text responses](#). In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 243–253.
- Omar Zaidan, Jason Eisner, and Christine Piatko. 2007. [Using “annotator rationales” to improve machine learning for text categorization](#). In *Human language technologies 2007: The conference of the North*

*American chapter of the association for computational linguistics; proceedings of the main conference*, pages 260–267.

Matthew D Zeiler and Rob Fergus. 2014. [Visualizing and understanding convolutional networks](#). In *European conference on computer vision*, pages 818–833. Springer.

Ye Zhang, Iain Marshall, and Byron C Wallace. 2016. [Rationale-augmented convolutional neural networks for text classification](#). In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2016, page 795. NIH Public Access.

Jiajie Zou, Yuran Zhang, Peiqing Jin, Cheng Luo, Xunyi Pan, and Nai Ding. 2021. [Palrace: Reading comprehension dataset with human data and labeled rationales](#). *arXiv preprint arXiv:2106.12373*.

## A Classification without fine-tuning

	Precision	Recall	F1-score
Label 0	0.49	0.46	0.47
Label 1	0.65	0.68	0.66
Macro Avg	0.57	0.57	0.57

Table 4: Overall classification performance metrics of SciBERT with no fine-tuning on question-answer pairs, where label 0 = Non-satisfactory and label 1 = Satisfactory.

# You've Got a Friend in ... a Language Model? A Comparison of Explanations of Multiple-Choice Items of Reading Comprehension between ChatGPT and Humans

George Dueñas<sup>1</sup>, Sergio Jimenez<sup>2</sup>, Geral Eduardo Mateus Ferro<sup>3</sup>

<sup>1</sup>Doctorado Interinstitucional en Educación, Universidad Pedagógica Nacional, Colombia

<sup>2</sup>Instituto Caro y Cuervo, Colombia

<sup>3</sup>Departamento de Lenguas, Universidad Pedagógica Nacional, Colombia

geduenasl@upn.edu.co, sergio.jimenez@caroycuervo.gov.co, gmateus@pedagogica.edu.co

## Abstract

Creating high-quality multiple-choice items requires careful attention to several factors, including ensuring that there is only one correct option, that options are independent of each other, that there is no overlap between options, and that each option is plausible. This attention is reflected in the explanations provided by human item-writers for each option. This study aimed to compare the creation of explanations of multiple-choice item options for reading comprehension by ChatGPT with those created by humans. We used two context-dependent multiple-choice item sets created based on Evidence-Centered Design. Results indicate that ChatGPT is capable of producing explanations with different type of information that are comparable to those created by humans. So that humans could benefit from additional information given to enhance their explanations. We conclude that ChatGPT ability to generate explanations for multiple-choice item options in reading comprehension tests is comparable to that of humans.

## 1 Introduction

Chatbots are used in education because they “promise to have a significant positive impact on learning success and student satisfaction” and “are promising tools to provide continuing feedback to lecturers and students” (Winkler and Söllner, 2018). According to Wollny et al. (2021), chatbots have been utilized in education to support learning and teaching, enhance services offered by educational institutions, promote well-being, and offer feedback and motivation. However, their use in assisting with the development of evaluation items, whether formative or summative, has not been widely explored.

The process of providing explanations for multiple-choice items can be more time-consuming and labor-intensive than constructing the item itself. This often results in the creation of numerous items

that lack explanations. Despite this, it is important to note that an item accompanied by explanations is significantly more versatile and useful than one without. Furthermore, the process of constructing explanations can reveal issues with the items that may not have been immediately apparent. As such, the implementation of a tool to assist item constructors in developing explanations could greatly enhance both the quantity and quality of items produced. Recent advancements in language models, such as ChatGPT, which have been trained on large amounts of text, show promise in their ability to assist with this task. This paper aims to investigate the efficacy of these models in comparison to explanations generated by humans.

## 2 Background

### 2.1 Explanations in multiple-choice items

Haladyna et al. (2002) proposed 31 multiple-choice item-writing guidelines focused on classroom assessment, but it can be applied to items used in other circumstances. They grouped these guidelines in five categories: Content concerns, Formatting concerns, Style concerns, Writing the stem, and Writing the choices. The last category is the most extensive one with 14 aspects, and in it, there are three aspects that are directly related to the explanation of the options: “Make sure that only one of these choices is the right answer”, “Keep choices independent”; choices should not be overlapping, and “Make all distractors plausible”. The provision of detailed explanations for each option is crucial in ensuring that only one option is unequivocally correct, while the remaining options contain inaccurate information that may appear plausible at first glance. On the other hand, they can be considered as a type of feedback (see Hattie, 2012, chap. 7) for students who require it, since the explanation for each option should include the reason why it is either correct or incorrect.

## 2.2 *Evaluar para Avanzar*

In Colombia, in 2020, the formative evaluation strategy called *Evaluar para Avanzar*<sup>1</sup> (EpA) was created by the Ministry of Education (MEN) and the ICFES (Colombian Institute for the Evaluation of Education). The aim of this strategy is to face the challenges of the COVID-19 pandemic by contributing to the classroom evaluations for students in grades 3 through 11 by means of complementary diagnostic instruments to the standardized tests. The assessment consists of two booklets per grade level for each academic year, each containing 20 items. For this study we selected the areas of Reading and Critical Reading for 5°, 9°, and 11° grades (see section 3.1). These selections correspond to the years 2021 and 2022 and were chosen because they contained only text-based items (MEN, 2006).

The framework used by ICFES to write these items is Evidence-Centered Design (ECD) (Mislevy et al., 2003, 2017). This means that the EpA items show information about the claims and evidences in the items. In this case, this type of information is the same as the *Saber*<sup>2</sup> standardized tests. The multiple-choice items in *Saber* 3°, 5°, 7°, and 9° have three *claims*<sup>3</sup>: (i) retrieve literal information expressed in fragments of the text, (ii) understand the local and global meaning of the text through inferences of implicit information, (iii) take a critical stance on the text by evaluating its form and content (Jurado and Rodríguez, 2020).

An example for the first claim (i) is item 1 of the 2022 grade 5 booklet, which reads as follows: *Según el texto, ¿en qué momento ocurre la historia?* [According to the text, when does the story take place?]

- A. Ocurre en este instante. [It happens right now.]
- \*B. Ocurrió en un tiempo lejano. [It happened in a distant time.]
- C. Ocurrió hace poco. [It happened recently.]
- D. Ocurrirá luego. [It will happen later.]

The test-taker must locate explicit information related to time that allows them to know when the

<sup>1</sup><https://www2.icfes.gov.co/en/caja-de-herramientas1>

<sup>2</sup><https://www.icfes.gov.co/es/web/guest/evaluaciones>

<sup>3</sup>It is a statement we'd like to be able to make about what a student knows or can do on the basis of observations in an assessment setting (Mislevy et al., 2003).

events described in the story occurred. The context begins with the expression “A long time ago...”, indicating that the events being described occurred in the distant past. The example in Table 2 is related to claim (ii), since the test-taker must deduce the meaning of a certain expression according to the given context.

In the case of *Saber* 11°, it also has three *claims*: (i) identify and understand the local contents that make up a text, (ii) understand how the parts of a text are articulated to give it a global meaning, (iii) reflect from a text and evaluate its content (Donoso, 2021). Therefore, each item contains the following information: a claim, an evidence, correct and incorrect options, as well as explanations for both the correct and incorrect options.

Regarding the difficulty of the items, they can be ranked in the following way: the items in the first claim would be *easy*, as they require retrieving explicit information from the contexts; the items in the second claim would be *intermediate*, as they require making inferences based on the information from the contexts; and finally, the items in the third claim would be *difficult*, as they require evaluating the information from the contexts.

## 2.3 ChatGPT on items and related

In November 2022, OpenAI released ChatGPT, which is a general-purpose language model “trained to follow an instruction in a prompt and provide a detailed response”<sup>4</sup>. This tool can write texts that are human-like and, at the same time, can “understand” the instructions it receives in some natural languages such as Spanish. So anyone, classroom teachers, learners and/or test developers, can use this artificial intelligence by creating an account at <https://chat.openai.com/>. Then all we have to do is ask questions and wait for this tool to generate answers and explanations, as simple as having a conversation. This methodology has been employed by researchers who have tasked ChatGPT with answering test items in order to evaluate its performance. Some examples come from areas such as law (Choi et al., 2023), medicine (Saraju et al., 2023; Gilson et al., 2023; Kung et al., 2023; Fijačko et al., 2023), among others (Guo et al., 2023; OpenAI, 2023).

Choi et al. (2023) used ChatGPT to produce answers to multiple choice and essay questions of four separate final exams for law school courses:

<sup>4</sup><https://openai.com/blog/chatgpt>

Constitutional Law; Federalism and Separation of Powers; Employee Benefits and Taxation; and Torts. The AI-generated answers were shuffled with student exams and graded blindly by three professors. They concluded that ChatGPT passed all exams and performed better on the essay components than on the multiple choice. [Sarraju et al. \(2023\)](#) created 25 open-ended questions addressing fundamental preventive concepts related to Cardiovascular Disease Prevention. Each question was given to ChatGPT 3 times and the responses were recorded. Then 3 reviewers graded each set of responses as “appropriate”, “inappropriate” or “unreliable”. ChatGPT gave 21 appropriate and 4 inappropriate answers. The authors observed great potential for interactive AI to assist clinical workflows (increased patient education and ease of patient-clinician communication), but these approaches must be further explored. [Fijačko et al. \(2023\)](#) given to ChatGPT 96 stand-alone and 30 scenario-based questions related to life support exams (BLS and ACLS). The authors concluded that although ChatGPT did not pass any of these exams, it has the potential to be a valuable resource for studying and preparing for life support exams.

In addition to evaluating the performance of ChatGPT, some studies have also assessed the explanations generated for its responses to test items. Our research aims to replicate this approach. [Gilson et al. \(2023\)](#) and [Kung et al. \(2023\)](#) analyzed the response explanations for user interpretability of the United States Medical Licensing Examination exams. The authors demonstrated that ChatGPT attained a score that is comparable to that of a third-year medical student and that its performance was either at or near the passing threshold for all the exams. Finally, [Guo et al. \(2023\)](#) evaluated, among other aspects, the answers created by ChatGPT with nearly 40K questions written in English and Chinese, and their corresponding answers created by human experts, creating the Human ChatGPT Comparison Corpus (HC3) dataset, coming from domains such as computer science, finance, medicine, law, psychology), and open-domain. The authors conclude that it is easier to distinguish the content generated by ChatGPT when an answer is provided for comparison than when the answer is provided alone and that those answers are considered more useful than those of humans.

Our objective is to compare the explanations generated by ChatGPT in Spanish for multiple-

choice reading comprehension items with those created by human item-writers (hereafter referred to as ‘humans’). As [Guo et al. \(2023, p. 2\)](#), we also want to know if ChatGPT can be “honest (not fabricate information or mislead the user), harmless (shouldn’t generate harmful or offensive content), and helpful (provide concrete and correct [item explanations])” to the humans.

We also want to evaluate whether ChatGPT can classify these items into one of the three *claims* used to build EpA, which are based on ECD.

In a similar way to the works of [Gilson et al. \(2023\)](#) and [Kung et al. \(2023\)](#), our work aims to provide qualitative and quantitative feedback on the performance of ChatGPT and assess its potential to help classroom teachers, learners, and test developers. To the best of our knowledge, no existing research has compared the explanations generated by ChatGPT with those created by humans for multiple-choice reading comprehension items. Given that our items span multiple school grades and text types in Spanish, we believe that this presents a unique and challenging opportunity to evaluate the capabilities of ChatGPT.

The rest of the paper is organized as follows. In Section 3 we provide a detailed description of the method and data used. In Section 4 we present and discuss the main results. Finally, in Section 5 we provide some conclusions and perspectives.

## 3 Methods

### 3.1 Data

The data consists of a set of human-written textual explanations for each of the item options from grades 5°, 9°, and 11° of the years 2021-2 and 2022-1 of EpA strategy respectively. We supplied ChatGPT with the context associated with the items and prompted it using natural language to generate responses and explanations for the corresponding items and options. The motivation for choosing these school grades is twofold: same grade level and consecutive booklets, but from different years, and each grade is the completion of a cycle in the Colombian educational system. In 5°, basic primary education is completed; in 9°, secondary basic education is completed, and in 11°, secondary education is completed. The latter is the one that allows a student to enter higher education.

Each grade level is accompanied by a booklet containing 20 items, which are organized into context-dependent groups of either 3 or 5 items per

context. Items that depend on an image context have been discarded, as ChatGPT does not support this particular format. Table 1 shows the different school grades, the types of contexts, the length of each context, and the respective assigned items. Thus, there are 46 items in 2021, and 42 in 2022, making a total of 88 items. Items grouped by claim (i.e., difficulty), year, and grade, according to ECD proposed by ICFES, are below. The underlined items were subsequently discarded due to the fact that the responses provided by ChatGPT did not align with the established answers in the booklets.

- **2021, 5°:** *claim 1:* 1, 2, 4, 8, 12; *claim 2:* 6, 9, 13, 15; *claim 3:* 3, 5, 7, 10, 11, 14.
- **2021, 9°:** *claim 1:* 7, 10, 11, 12; *claim 2:* 8, 9, 13, 15, 18; *claim 3:* 6, 14, 16, 17, 19, 20.
- **2021, 11°:** *claim 1:* 8, 14, 16, 19; *claim 2:* 1, 2, 3, 6, 13, 18, 20; *claim 3:* 4, 5, 7, 15, 17.
- **2022, 5°:** *claim 1:* 1, 2, 5, 9, 17; *claim 2:* 3, 4, 6, 10, 11, 12, 18; *claim 3:* 7, 8, 19, 20.
- **2022, 9°:** *claim 1:* 1, 2, 4, 11; *claim 2:* 3, 5, 12, 13, 14; *claim 3:* 15.
- **2022, 11°:** *claim 1:* 2, 3, 13, 18; *claim 2:* 8, 9, 10, 15, 16, 17, 19; *claim 3:* 1, 11, 12, 14, 20.

### 3.2 Data Extraction

As the booklets with the contexts and items are public and available in PDF files, we manually copied each context and its respective items, and pasted them into a plain text file. Subsequently, we checked that the texts matched, since sometimes the texts copied from the PDF file pasted with errors in some characters. Afterwards, the explanation for each incorrect option was manually extracted, since these explanations were grouped together in a single paragraph. When the paragraph began with the following statement: “The options X, Y, and Z are not correct, because...” (where X, Y, and Z represent the incorrect options), that part was added to each of the explanations of the three options. The motivation behind this was to expand the explanation created by humans to compare each of these explanations with its corresponding one created by ChatGPT. Finally, this same information was pasted back into a spreadsheet<sup>5</sup>, where the

<sup>5</sup><https://docs.google.com/spreadsheets/d/1CTXEJn0dT-4xzUYrwZZJDPMe-XnyvAHivYgPxG4ejCY/edit?usp=sharing>

information created by ChatGPT was also added.

As to ChatGPT, we collected explanations for each option between December 2022 and January 2023. In those months, ChatGPT was only available through its website, so we collected the information as follows:

1. The item context is copied and pasted into the input box, and the explanation given was omitted. We used a chat session for each context and its respective items so that the ChatGPT memory retention function only takes into account context-related and item-related information. This was done because, like humans, when constructing items, they take into account what has been said in the context and in the other items in order to fulfill what Haladyna et al. (2002) have outlined.
2. The first multiple-choice item is copied and pasted into the input box. The answer and the explanation are saved.
3. ChatGPT is asked the following: “In which of the following categories would the above question be classified?”, where the categories are the *claims* used to build EpA. The answer and the explanation are saved.
4. Step 2 is performed again<sup>6</sup> to subsequently ask ChatGPT the following: “Why is option X incorrect?”, where X corresponds to each of the incorrect options of the respective item.
5. The next multiple-choice item is copied and pasted into the input box. The answer and the explanation are saved.

### 3.3 Comparing explanations

Given that the explanations authored by humans have been publicly available since 2021, no experiment was conducted to ascertain which explanation - human-authored or ChatGPT-generated - was deemed more suitable by teachers or other individuals. Instead, for each pair of explanations per option for the items, a manual review was carried out to identify the differences and similarities between them as long as ChatGPT selects the correct option (the key) according to the one established in

<sup>6</sup>This step must be done again because ChatGPT is susceptible to the immediately preceding text, so if asked why X option is incorrect, its response will be based on the question of the three categories. ChatGPT again selected the same option for each item.

2021-2					2022-1				
Gr	Text Type	Words	Items	Total	Gr	Text Type	Words	Items	Total
5	expository	274	1-5		5	narrative	296	1-4	
5	narrative	337	6-10	15	5	descriptive	311	5-8	16
5	expository	344	11-15		5	expository	225	9-12	
9	narrative	373	6-11		5	expository	203	17-20	
9	descriptive	108	12-15	15	9	narrative	208	1-5	10
9	descriptive	291	16-20		9	narrative	269	11-15	
11	expository	392	1-4		11	narrative	490	1-3	
11	argumentative	180	5-8	16	11	argumentative	264	8-11	16
11	narrative	143	13-16			11	argumentative	302	
11	narrative	420	17-20		11	argumentative	512	17-20	
				46					42

Table 1: Types of (con)texts by grade and year, length of each (con)text, and their respective assigned items. Gr stand for the school grade.

the booklet. The review was performed by the first author of this study, a linguist and native Spanish speaker with experience in writing reading comprehension items for national tests in Colombia.

This review involves a comparative analysis of each pair of explanations associated with each item option, irrespective of whether the option is correct or incorrect. To do that, six colored tags have been created to annotate the differences and similarities between the explanations. The explanations of the underlined items above were not compared because the answer given by ChatGPT did not match with that established in the booklets. The first tag, denoted as MIC and colored red, highlights text passages that explain the respective option and where there is agreement between humans and ChatGPT - that is, the passages convey the same meaning despite being phrased differently.

The second tag (AII) is colored green. This text has been created by humans and it is generally used to refer to part or all of the context expanding its explanation or providing additional information (such as the function of words, punctuation marks, or titles in a text). The meaning of this text does not have a match with any part of the explanation created by ChatGPT.

The third tag (AIC) is colored blue. This text has been created by ChatGPT and it is generally used to refer to part or all of the context expanding its explanation or providing additional information (such as the function of words, punctuation marks, or titles in a text). The meaning of this text does not have a match with any part of the explanation created by humans.

The fourth tag (EIC) is colored gray, where the texts highlighted intend to expand the explanations by making use of or referring to some part of the Context. In this texts, humans and ChatGPT have matched, that is, they have the same meaning although written differently.

The fifth tag (CI) is colored brown. This text has been created by humans to close the option explanation in an individually or generally (when it is the last option) way. One way to identify the closure is by using connectors such as *therefore*, *thus*, or *so* (*por lo tanto* or *por tanto* in Spanish).

The sixth tag (CC) is colored fuchsia. This text has been created by ChatGPT to close the option explanation. In the case of ChatGPT, it adds or repeats the reason why the option is (in)correct, but with other words. One way to identify the closure is by using connectors such as *therefore*, *thus*, or *so* (*por lo tanto* or *por tanto* in Spanish).

Table 2 shows an example of the explanations labeled by option from their comparison for item 6 of grade 5 of 2022-1 including a Venn diagram for visualizing color tags.

## 4 Results and Discussion

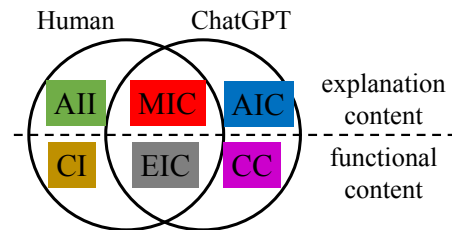
We utilized a traditional test approach of simply presenting the item to ChatGPT. Out of the total 88 items, ChatGPT answered 78 correctly, resulting in the exclusion of 10 items. Table 3 shows the average length of explanations in number of words per color tag for correct and incorrect answers, by school grade, and by humans and ChatGPT. It also includes effect sizes and p-values of the differences.

An analysis of the common information pro-



What does the author refer to when using the word “wonders” in the last paragraph of the text?

- A. That the tools were very old.
- \*B. To the ability of hunters to build effective tools.
- C. To the ease with which hunters caught mastodons.
- D. That the tools were very beautiful.



Human explanation	ChatGPT explanation
<p><b>A.</b> The attribution of wonders is not given to the tools for their antiquity. The value is given to the ability of the hunters and the effectiveness of the tools, for this reason, the options A, C and D are not valid.</p>	<p>The option A, “That the tools were ‘very old’”, is incorrect as an answer to the item ... In the last paragraph of the text, the author mentions that ... These tools were “wonders”, suggesting they were very good or effective ... Instead, the option A suggests that the tools were very old, which ... Therefore, the option A is incorrect as an answer to the item.</p>
<p><b>*B.</b> In this case, the item requires students to deduce what the text attributes the character of “wonders” in the text, according to the statement ... that appears at the end of the news. The semantic feature of wonders does not refer only to the tools but also to the ability of hunters to build effective tools, therefore, the correct option is B.</p>	<p>The author refers to the ability of hunters to build effective tools when using the word “wonders” in the last paragraph of the text. In the last paragraph of the text, the author mentions that ... These tools were “wonders”, suggesting that they were very good or effective ... Therefore, it is possible to infer that the author uses the word “wonders” ...</p>
<p><b>C.</b> The attribution of wonders is not given to the tools for the ease with which hunters caught mastodons. The value is given to the ability of the hunters and the effectiveness of the tools, for this reason, the options A, C y D are not valid.</p>	<p>The option C, “To the ease with which hunters caught mastodons”, is incorrect as an answer to the item ... In the last paragraph of the text, the author mentions that ... These tools were “wonders”, suggesting that they were very good or effective ... Instead, the option C suggests that hunters caught mastodons easily, which ... Therefore, the option C is incorrect as an answer to the item.</p>
<p><b>D.</b> The attribution of wonders is not given to the tools because the tools were beautiful. The value is given to the ability of the hunters and the effectiveness of the tools, therefore, the options A, C y D are not valid.</p>	<p>The option D, “That the tools were very beautiful”, is incorrect as an answer ... In the last paragraph of the text, the author mentions that ... These tools were “wonders”, suggesting that they were very good or effective ... Instead, the option D suggests that the tools were very beautiful, which ... Therefore, the option D is incorrect as an answer to the item.</p>

Table 2: Tagging example of the explanations for item 6 of 5th grade in 2022-1 (texts translated from Spanish).

vided in explanations by humans and ChatGPT (MIC and EIC tags) reveals that ChatGPT’s writing tends to be more verbose than that of humans in all scenarios except for 9th grade in the MIC tag. The test results presented in the penultimate row of Table 3 indicate a significant difference in

the length of prose between humans and ChatGPT when expressing the same content. However, the effect size of this difference is small. With respect to these differences in additional explanations (last row), ChatGPT texts are considerably larger than those of humans, exhibiting a large effect size.

The length of the explanations can be attributed to two factors: firstly, humans tend to be direct in their responses and may not offer additional information, as evident from the minimal or non-existent explanations provided by 9th and 11th grades for the AII tag. Furthermore, the results for the CI tag suggest that humans do not provide closure explanations. Secondly, ChatGPT explanations tend to be longer due to various factors, such as repetition the option, quoting fragments of the context, reiterating the status of the option (correct or incorrect) or all of the above. Thus, ChatGPT performed information expansion for almost half of the explanations (as seen in the AIC tag) and added a closure for one-fifth of the explanations.

When comparing the length of explanations for correct vs. incorrect options the only significant differences were observed in AII, AIC and CI tags. This let us conclude that humans prefer to provide additional explanations to correct options, while ChatGPT does it to incorrect options, both with a medium effect size in the difference of their respective preferences. Similarly, for the functional content, humans preserve such preference, while ChatGPT do the same but not significantly.

When comparing the length of explanations for correct versus incorrect options, significant differences were observed only in the AII, AIC, and CI tags. This allows us to conclude that humans prefer to provide additional explanations for correct options, while ChatGPT does so for incorrect options, both with a medium effect size in the difference of their respective preferences. Similarly, for functional content, humans maintain this preference, while ChatGPT does the same but not significantly. More importantly, both humans and ChatGPT agree on not having differences in the length of the main common explanations for correct or incorrect options.

Another important factor is the ability of ChatGPT of identifying the correct option. Among the 31 items to grade 5, ChatGPT fails to provide a correct answer for item 20 in the 2022 dataset. In its explanation for the four options, ChatGPT notes that “There has been no mention of option X being wrong at any point”, and gives an explanation for each option. Although ChatGPT did not choose the correct option for this item, its explanations were compared to determine their accuracy. ChatGPT considered that each incorrect option (distractor) could be correct. However, upon comparing these

explanations to those created by humans, it was determined that ChatGPT’s explanations were incorrect. A more extensive discussion of these results exceeds the scope of this paper and is left for future research as the information is available in the spreadsheet. As a result, this item was deemed invalid and excluded from the analysis.

For the remaining 30 items, it can be seen that the explanation given by humans and ChatGPT have matched in the meaning, but with different words (MIC). Regarding the tags AII and AIC, it is evident that AIC is more commonly used. This suggests that ChatGPT often includes additional information to provide a more comprehensive explanation, based on either a specific portion or the entire context. Something similar occurs with the CI and CC tags, where the latter occurs slightly more frequently than the former. It is evident that approximately one-third of the explanations for the correct options have a concluding statement provided by both humans and ChatGPT. Finally, regarding the EIC label, humans and ChatGPT explanations rarely coincide and expand by referencing parts of the context. Similar to the previous findings, the comparison of explanations by grade, tag, and correctness reveals that ChatGPT tends to provide more information than humans, albeit not uniformly for all options. Another factor in which ChatGPT may fail is its inability to affirm that other options are incorrect, even when it has correctly chosen the correct option. In item 14 of grade 5 from 2021, ChatGPT indicates that “Option A is not incorrect” and provides an explanation to support this assertion.

Among the 25 items to grade 9, ChatGPT fails to provide a correct answer for item 15 in the 2021 and item 1 in the 2022. Regarding item 15, ChatGPT selected option B, which is incorrect. This may be because ChatGPT omitted or confused some words. The text states that “Simone reflects on her own life as a woman and after this reflection, publishes the book *The Second Sex*”, but as the human who built this item says in his/her explanation: “[said quote] is not a reflection on sex”. Regarding item 1, ChatGPT did not select any of the four options as correct and in its explanations it stated that the word whose meaning is contrary to the word “illegal” is “legal”, thus omitting the correct option: “allowed”. This may be because ChatGPT discarded the presented context in which the word “allowed” (permitted) fits as the semantic opposite

Grade	Option	Explanation content				Functional content			
		Human		ChatGPT		Human		ChatGPT	
		AII	MIC	MIC	AIC	CI	EIC	EIC	CC
5°	correct	10(17)	38(22)	46(22)	39(28)	3(5)	1(3)	2(6)	7(10)
5°	incorrect	4(11)	37(14)	46(21)	43(34)	1(3)	1(3)	6(3)	9(11)
9°	correct	2(7)	59(22)	56(25)	20(24)	0(0)	1(7)	3(14)	17(23)
9°	incorrect	0(0)	62(16)	54(18)	23(28)	0(8)	0(0)	0(0)	10(9)
11°	correct	1(3)	48(23)	62(28)	15(21)	0(0)	3(10)	3(12)	18(13)
11°	incorrect	3(9)	51(15)	53(24)	34(25)	0(3)	1(3)	1(5)	9(10)
average	correct	5(12)	48(24)	54(25)	25(27)	1(3)	2(7)	3(11)	13(16)
average	incorrect	3(9)	49(18)	51(21)	34(30)	0(2)	0(2)	0(3)	9(10)
	Effect size correct vs. incorrect†	0.456	0.547	0.460	0.577	0.463	0.481	0.481	0.483
	p-value	<b>0.034</b>	0.222	0.291	<b>0.039</b>	<b>0.017</b>	0.126	0.120	0.089
average	both	3(10)	49(20)	52(22)	32(30)	1(3)	1(4)	1(6)	10(12)
	Effect size of diff. [p-value]‡‡		0.111	<b>[0.047]</b>			0.170	<b>[0.004]</b>	
	Effect size of diff. [p-value]‡‡		0.717	<b>[&lt;.001]</b>			0.642	<b>[&lt;.001]</b>	

†Effect size for Wilcoxon test calculated as  $\frac{z}{\sqrt{N}}$

‡‡Effect size for Mann-Whitney test calculated as  $\frac{U}{n_1 \times n_2}$

Table 3: Average (STD) number of words per color tag (significant differences having  $p < 0.05$  showed in boldface).

of the word “illegal”. In item 20 of grade 9 of 2021, ChatGPT states for options A and C that “There is not enough information in the text to determine if the option . . . is incorrect or not. The text does not explicitly mention who it is intended for”.

Among the 36 items to grade 11, ChatGPT fails to provide a correct answer for 5 items of 2021 (1, 14, 16, 17, 20) and 2 items of 2022 (12, 20). In item 1, ChatGPT selected option A, which is incorrect. This option asserts that the argumentative relationship between the two presented statements is one of premise and evidence, while the correct option provided by the human is D, which asserts that the relationship is conjecture and counterevidence. Regarding the remaining items, items 14 and 16 belong to the same context, and their responses are derived from the same fragment. Furthermore, these items ask for information that is explicit in the text (claim 1), making it uncommon for ChatGPT to provide incorrect answers. Similarly, items 17 and 20 are associated with the same context (but different from before), although their answers are derived from different fragments, and they pertain to different claims (2 and 3, respectively). In general terms, ChatGPT could have provided incorrect responses, but the comparison of these types of explanations precisely calls for a more in-depth analysis, which falls outside the scope of this work.

In three items of the grade 11, ChatGPT correctly

identified the correct option but did not provide a conclusive explanation for why the other options were incorrect. In item 13 of 2021, ChatGPT indicates that “Option X is not incorrect” and adds that “this option does seem to be a reason for the character’s feeling of unease”. This type of explanation may be due to the format of the question, which uses a negation structure: “Which NOT”. For items 10 and 11 of 2022, which are based on a fragment of HAMLET’S MONOLOGUE, ChatGPT provided explanations that diverged from the expected pattern. For item 10, ChatGPT stated that both options A and C “is not incorrect, but rather one of the options that can adequately describe the above text”. The explanation then goes on to clarify why the adjective “philosophical” (option A) or “poetic” (option C) would be a better fit for the fragment. In item 11, something similar to item 13 of 2021 occurs, where ChatGPT selected the correct option, but regarding the incorrect options, it indicated that “it is not incorrect, but rather it is a statement that Hamlet mentions and reflects on in his monologue”. The question for this item also had the “Which NOT” structure: “Based on the above text, which of the following statements would Hamlet NOT agree with?”. It is worth noting that in ChatGPT’s explanation for the correct option, it also states that options A, B, and D are incorrect, since it states: “The other statements

(A, B and D) do seem to agree with the content of Hamlet’s monologue”. Given the above, only the explanation for the correct option was compared, while the explanations for the remaining incorrect options were not evaluated. Due to space constraints, we cannot fully explore the analysis of these differences in this study.

Regarding the classification of the items in the three claims, ChatGPT correctly classified 49 (55.68 %) items into the three claims. Below are the items that were classified in a wrong claim.

- **2021, 5<sup>o</sup>:** *claim 1:* 6, 11, 15; *claim 2:* 3, 5, 7, 10.
- **2021, 9<sup>o</sup>:** *claim 1:* 6, 8, 9, 14, 15, 16; *claim 2:* 19, 20.
- **2021, 11<sup>o</sup>:** *claim 1:* 4, 6, 13; *claim 3:* 2, 3, 18.
- **2022, 5<sup>o</sup>:** *claim 1:* 3, 4, 7, 11, 12; *claim 2:* 8, 19.
- **2022, 9<sup>o</sup>:** *claim 1:* 12, 15; *claim 3:* 13.
- **2022, 11<sup>o</sup>:** *claim 1:* 8, 10, 14, 17; *claim 2:* 1, 11, 12, 20.

Among the items of the grade 5, 14 were classified incorrectly. Items 3, 5, 7, 10, 18, and 19 were reclassified from claim 3 to claim 2, while items 3, 4, 6, 11, 12, and 15 were reclassified from claim 2 to claim 1. Additionally, items 7 and 11 were reclassified from claim 3 to claim 1. Something similar occurs with grades 9 and 11, where in the former 11 items were misclassified, while in the latter 14 items were misclassified. We hypothesize that ChatGPT misclassifies certain items from claim 3 due to the presence of quoted expressions from the context in the question. It appears that ChatGPT interprets these quotes as literal text and consequently categorizes them under claim 1, but we are unable to delve further into the analysis of this classification by ChatGPT in this study and further investigation is needed.

## 5 Conclusion and Future Work

This study provides insights into the creation of explanations for multiple-choice item options in reading comprehension tests with the assistance of AI. By comparing explanations generated by ChatGPT with those created by humans, our analysis indicates that ChatGPT can produce explanations that could be considered equivalent and possibly

better than those created by humans, with potential benefits for both humans and language models. ChatGPT can offer more detailed and specific insights into the text, which can enhance the quality of explanations provided by humans. However, our findings also suggest that there is still room for improvement for both humans and language models. To address these limitations, future research could explore ways to combine the strengths of humans and language models to produce even more accurate and informative explanations. Therefore, ChatGPT has the potential to assist teachers and other professionals in the creation of high-quality assessment items through a well-designed prompt, which can help ensure that items have a single correct answer, independent options, non-overlapping options, and plausible options. Furthermore, ChatGPT ability to classify items based on ECD principles is promising, but further research is needed. For example, the evidences could be provided to language models and ask them to classify each item in one of them. Also, they could be asked to create the options and the respective explanations based on some kind of guidelines such as the one cited.

## References

- Jonathan H Choi, Kristin E Hickman, Amy Monahan, and Daniel Schwarcz. 2023. [Chatgpt goes to law school](#). *SSRN. Minnesota Legal Studies Research Paper*, (23-03).
- Lizeth Donoso. 2021. *Prueba de Lectura Crítica Saber 11.º, Saber TyT y Saber Pro. Marco de referencia para la evaluación*. Instituto Colombiano para la Evaluación de la Educación (ICFES).
- Nino Fijačko, Lucija Gosak, Gregor Štiglic, Christopher T Picard, and Matthew John Douma. 2023. [Can chatgpt pass the life support exams without entering the american heart association course?](#) *Resuscitation*, 185.
- Aidan Gilson, Conrad W Safranek, Thomas Huang, Vimig Socrates, Ling Chi, Richard Andrew Taylor, and David Chartash. 2023. [How does chatgpt perform on the united states medical licensing examination? the implications of large language models for medical education and knowledge assessment](#). *JMIR Med Educ*, 9:e45312.
- Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. [How close is chatgpt to human experts? comparison corpus, evaluation, and detection](#).
- Thomas M Haladyna, Steven M Downing, and Michael C Rodriguez. 2002. [A review of multiple-](#)

- choice item-writing guidelines for classroom assessment. *Applied measurement in education*, 15(3):309–333.
- John Hattie. 2012. *Visible learning for teachers: Maximizing impact on learning*, chapter The flow of the lesson: the place of feedback. Routledge.
- Fabio de Jesús Jurado and María Elvira Rodríguez. 2020. *Competencias Comunicativas en Lenguaje: Lectura y Escritura. Marco de referencia para la evaluación*. Instituto Colombiano para la Evaluación de la Educación (ICFES).
- Tiffany H Kung, Morgan Cheatham, Arielle Medenilla, Czarina Sillos, Lorie De Leon, Camille Elepaño, Maria Madriaga, Rimel Aggabao, Giezel Diaz-Candido, James Maningo, et al. 2023. Performance of chatgpt on usmle: Potential for ai-assisted medical education using large language models. *PLOS Digital Health*, 2(2):e0000198.
- MEN. 2006. *Estándares Básicos de Competencias en Lenguaje, Matemáticas, Ciencias y Ciudadanas. Guía sobre lo que los estudiantes deben saber y saber hacer con lo que aprenden*. Ministerio de Educación Nacional, Colombia.
- Robert J Mislevy, Geneva Haertel, Michelle Riconscente, Daisy Wise Rutstein, and Cindy Ziker. 2017. *Assessing Model-Based Reasoning using Evidence-Centered Design: A Suite of Research-Based Design Patterns*, chapter Evidence-centered assessment design. Springer.
- Robert J Mislevy, Linda S Steinberg, and Russell G Almond. 2003. On the structure of educational assessments. *Measurement: Interdisciplinary research and perspectives*, 1(1):3–62.
- OpenAI. 2023. Gpt-4 technical report.
- Ashish Sarraju, Dennis Bruemmer, Erik Van Iterson, Leslie Cho, Fatima Rodriguez, and Luke Laffin. 2023. Appropriateness of cardiovascular disease prevention recommendations obtained from a popular online chat-based artificial intelligence model. *JAMA*, 329(10):842–844.
- Rainer Winkler and Matthias Söllner. 2018. Unleashing the potential of chatbots in education: A state-of-the-art analysis. In *Academy of Management Annual Meeting*.
- Sebastian Wollny, Jan Schneider, Daniele Di Mitri, Joshua Weidlich, Marc Rittberger, and Hendrik Drachler. 2021. Are we there yet? - a systematic literature review on chatbots in education. *Frontiers in artificial intelligence*, 4:654924.

# Automatically Generated Summaries of Video Lectures May Enhance Students' Learning Experience

Hannah Gonzalez\*, Jiening Li\*, Helen Jin\*,  
Jiaxuan Ren\*, Hongyu Zhang\*, Ayotomiwa Akinyele\*,  
Adrian Wang, Eleni Miltsakaki, Ryan S. Baker, Chris Callison-Burch  
University of Pennsylvania

`hannahgl, jiening, helenjin, rjx, hz53, tomiwa, kelmp, elenimi, rybaker, ccb@seas.upenn.edu`

## Abstract

We introduce a novel technique for automatically summarizing lecture videos using large language models such as GPT-3 and we present a user study investigating the effects on the studying experience when automatic summaries are added to lecture videos. We test students under different conditions and find that the students who are shown a summary next to a lecture video perform better on quizzes designed to test the course materials than the students who have access only to the video or the summary. Our findings suggest that adding automatic summaries to lecture videos enhances the learning experience. Qualitatively, students preferred summaries when studying under time constraints.

## 1 Introduction

Video lectures have been an important part of scaled online courses and flipped classrooms for several years, and have become widely used for an increasingly larger range of courses as a substitute for students unable to attend class due to the COVID-19 pandemic (van Alten et al., 2020). Past research in human-computer interaction aimed to improve educational videos via interactive transcripts, word clouds, keyword search, and highlight storyboards (Kim et al., 2014), or by segmenting the videos to present highlight moments with snapshots and transcripts (Yang et al., 2022). Others have created video digests that are organized into a textbook-like format with chapters, titles, and sections with text summaries (Pavel et al., 2014). Pavel et al. (2014)'s system provides an authoring interface that allows video authors to manually write textual summaries of a video themselves or to send the video to a crowdsourcing service to have summaries written. Textual summaries are believed to be effective in helping students review course materials. For example, Shimada et al. (2017) find

that students using summaries of slides for preview have higher pre-quiz scores and spend less time, compared to students previewing original learning materials.

In this work, we investigate the feasibility of automatically summarizing lecture videos' transcripts using recent advances in large language models such as GPT-3 (Brown et al., 2020). We are encouraged by recent research in natural language processing demonstrating that people often prefer GPT-3 generated summaries over other methods of automatically generated summaries for news (Goyal et al., 2022).

The availability of high-quality automatic summaries would allow their use in a wide range of online courses. In this paper, we first detail our method for creating an automatic summarizer of video lectures. Then, we report a controlled user study that we conducted with around 100 university students to investigate whether the automatic summaries were an effective study aid.

## 2 Automatic Video Lecture Summarizer

We designed an automatic lecture summarizer that leverages recent advances in large language models (LLMs). Large language models are trained on massive language datasets, thus creating rich representations of language content. These models have been especially successful in generating fluent responses in a variety of natural language generation tasks with zero or few-shot prompting (Brown et al., 2020; Liu et al., 2023), instruction following (Sanh et al., 2021; Mishra et al., 2022; Bach et al., 2022), or fine-tuning (Howard and Ruder, 2018; Wei et al., 2021).

It is well-known, albeit not well-understood, that the quality of GPT-3 language generation is affected by the given prompt even if the differences in the way the prompt is articulated are not semantically very different. It is, also, unclear for which tasks it is sufficient to give prompts to achieve suc-

\* These authors contributed equally.

successful GPT-3 responses or whether there is a need for fine-tuning a model. For these reasons, we experimented with three approaches to generating lecture summaries.

In all of our experiments, the input provided to the model is retrieved by transcribing video lectures. For the video transcription, we use Amazon Transcribe. The transcribed video content is then split into chunks of a maximum of 4,000 tokens. This is the maximum length that can be encoded by GPT-3. The final summary of the lecture is the aggregate of the summaries of its parts. For each chunk, we retain the corresponding start time in the video.

**(1) Zero prompt summarization.** In the first approach, for each retrieved chunk we generated a summary providing the following prompt:

```
I am a summarization bot. If you give me text, I
will provide a textbook-like summary.
Text: <transcript_chunk>
Summary:
```

In this condition, the summary is generated without any training. We call this approach the "zero prompt" condition. Below we provide an input example of a raw transcript from a Natural Language Processing lecture given in an Artificial Intelligence university course.

#### Sample input to summarizer:

```
So uh if we're just talking about, we were just
talking about input sequences for our problem
with dimensionality, then let's imagine that we've
got an input sequence of like 10 And we've got
of okay, let's imagine we have a vocabulary size
of 100,000. then we've got 100,000 to the power
of 10 different possible sequences, word combi-
nations. So that ended of itself is going to be 10
to the power of 50 parameters like unimaginably
large, way more then? The number of ida adam
is in the universe? So too high to estimate from
data.
```

#### Summarizer output:

```
If we have a vocabulary size of 100,000 and an
input sequence of 10, there are 100,000 to the
power of 10 different possible sequences. This is
too high to estimate from data.
```

As can be seen in the summarizer output above, the zero-shot summarizer removes filled pauses like *um* and *uh*, eliminates false starts like the repetition of *let's imagine*, deletes misrecognized phrases like *the number of ida adam* (which should be *the number of atoms*), and concisely summarizes the idea in the transcript chunk.

In Appendix A we provide details about the model variant and parameters that we used.

**(2) Few shot prompt summarization.** In the second approach, we enriched the prompt by providing ten prompt-summary pairs as context to the current prompt.

In this approach, we provide a few training examples to the GPT-3 model and we call it the "few shot prompt" condition. The structure of this few-shot prompting is as follows:

```
I am a summarization bot. If you give me text, I
will provide a textbook-like summary without re-
peating past summaries or describing the speaker.
Text: <chunk from 10 prompts ago>
Summary: <summary of that chunk>
...
Text: <chunk from 9 prompts ago>
Summary: <summary of that chunk>
...
Text: <chunk from 1 prompt ago>
Summary: <summary of that chunk>
Text: <current_transcript_chunk>
Summary:
```

In this condition, we modified the prompt by adding to it "without repeating past summaries or describing the speaker." This modification prevented the summarizer from a) repeating past summaries and b) starting summaries with statements like *The speaker is discussing [topic]*. To further reduce repetition errors, we included in the summarizer a step to check if the current summary output matches any of the previous summaries. If so, GPT-3 would be prompted to generate a new output on the same prompt.

We observed that there were several advantages to including previous chunks and their summaries as part of the input. First, they provide useful context for subsequent summaries to remain topical. Second, transcription errors are not always uniform across chunks. For example, the term *n-gram* is misrecognized in the following chunk of the transcript.

```
What we what we're doing is basically just con-
structing a table. Look I wanted to say here's
a sequence. What's the probability of the next
word? Just looking up at the table. And the trick
for these engram based language models was how
do we deal with unseen sequences? So how do
we deal with new combinations of n words that
were never that never occurred in our training So
we did things like smoothing, we did things like
interpretation. We did back off too small, the two
smaller and smaller sequences.
```

Due to the context given previously, the summarizer provides a correction in its output, as can be

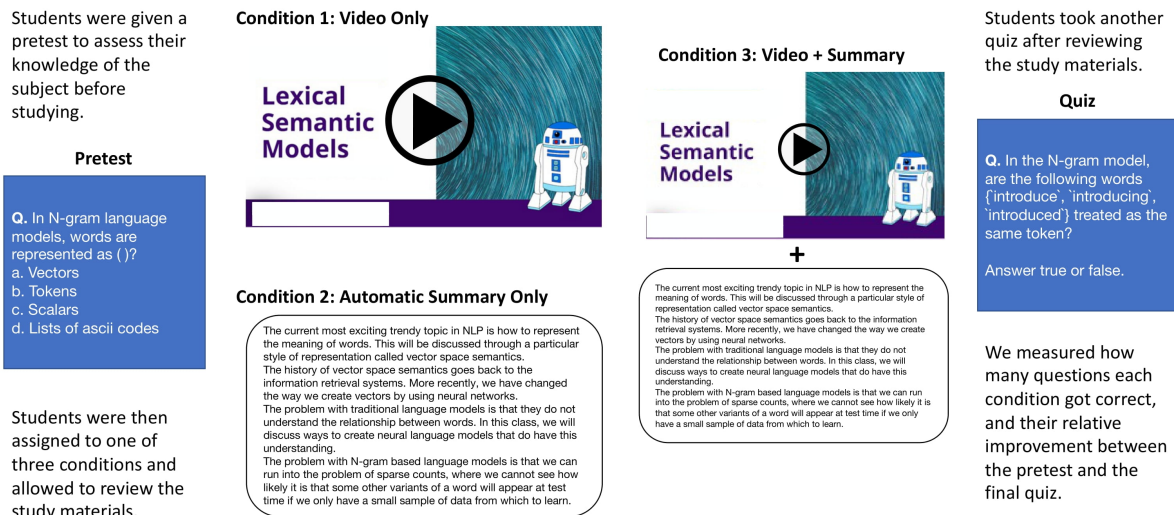


Figure 1: We introduce a new method for creating automatic summaries of lecture video transcripts, and perform a user study with 106 students to determine whether supplementing videos with the summaries enhances their learning.

seen in the output shown below (but not for the word *interpolation* that remains incorrectly transcribed as *interpretation*):

The trick for **n-gram** based language models is how to deal with unseen sequences. This is done with smoothing, interpretation, and back off to smaller sequences.

Table 2 in the Appendix gives more examples of the automatic summaries produced by our few-shot model.

**(3) Fine-tuning GPT-3.** In the third approach, we experimented with fine-tuning GPT-3 to perform lecture summarization. We manually edited the output of our few-shot model described above in order to provide annotated examples for fine-tuning. By this process, we obtained 114 prompt/output pairs which we then used to fine-tune a summarization model. When fine-tuning a GPT-3 model, we no longer need to provide prompts like we did in the previous two approaches.

The motivation behind experimenting with different approaches to summarizing transcribed video lectures was to identify a model that is likely to yield quality summaries. Through a series of informal evaluations of the three types of outputs, we observed that the fine-tuned model produced summaries that were more consistent in style and contained less repetition than the zero-shot and few-shot models. Table 3 in the Appendix gives examples of the automatic summaries produced by our fine-tuned model. As our main interest in this study is to evaluate whether adding summaries to

video lectures yields learning benefits to students' review of course materials, we did not perform a formal evaluation of the three approaches to automatic summarization. Instead, we opted to conduct a controlled study to evaluate the learning benefits of summarization in three course reviewing conditions. We report this evaluation study in the next section.

### 3 Evaluation Study

In this section, we report a controlled study that we conducted with the goal of evaluating the potential benefits of offering students an automatic summary of transcribed video lectures. In what follows, we describe the participants of the study, the testing conditions, and the results.

**Participants.** We recruited 106 undergraduate and Master's students who were taking an Artificial Intelligence course in Fall 2022. Students were given extra course credit for their participation.

**Study design.** We evaluated student performance on materials that students reviewed for two upcoming topics in the course presented in video lectures. These consisted of two short, pre-recorded lecture videos on Lexical Semantic Models (10 minutes) and Stochastic Gradient Descent (12 minutes). For each topic, we evaluated the three learning conditions listed below.

#### Testing conditions

1) Reviewing only the lecture video, 2) Reviewing only the automatically generated summary, 3) Reviewing both the video and the automatic summary.



Cond.	N	Pre-test	$\sigma$	Post-test	$\sigma$	$\Delta$
Sum.	56	62%	1.03	73%	1.4	17.7%
Video	39	67%	0.9	79.7%	1.1	18.5%
V+S	48	66%	0.9	82%	1.2	24.4%

Table 1: Mean correctness on the pre-test quizzes, and mean correctness on the quiz after reviewing the study materials for students in each of our three learning conditions for both lectures: Summary, Video, and Both.

All students were randomly assigned to a different learning condition for each topic. Many participants reviewed both lecture topics. For the second round, they were assigned to a different learning condition.

Prior to reviewing the course materials, students were given pre-test quizzes with four questions for each topic to test their initial understanding of the concepts. The answers were not shown to the students. After the pre-test, students reviewed the course materials using the materials associated with their randomly assigned learning condition. Finally, they answered a 10-question quiz on the material that they had reviewed. These included the 4 questions from the pre-test, plus 6 previously unseen questions. The quiz questions consisted of a mix of True/False questions and multiple-choice questions. The quiz questions are given in Appendix D.

The students were not given a time constraint for reviewing the materials. However, once they started the quiz, they were no longer allowed to review the materials. Table 1 summarizes the students' performance under the different learning conditions. We calculated the relative percentage point increase as follows:

$$\Delta = \frac{\text{Post score (\%)} - \text{Pre score (\%)}}{\text{Pre score (\%)}}$$

The mean correctness on the pre-test quizzes is below 70%. After reviewing the learning materials, the condition in which students demonstrate the smallest improvement is condition (2) with only access to the automatic summaries improve: a relative improvement of 17.7% or an 11% absolute improvement. Students who reviewed only the videos (condition 1) have a relative improvement of 18.5% (12.7% absolute). Students who reviewed both the videos and the automatic summaries have a relative improvement of 24.4% (16% absolute). A finer-grained breakdown of the students' performance on the quizzes for each lecture video is given in Appendix F.

We conducted a paired t-test to determine if there

was a significant difference in the test correctness scores before and after the video+summary intervention. The results showed a calculated t-statistic of 2.12 and a p-value of 0.045 for the Lexical Semantic Lecture, as well as a calculated t-statistic of -4.16 and a p-value of 0.0003 for the Stochastic Gradient Descent Lecture. These findings indicate a significant difference between the means. Although we cannot conclusively determine that the video+summary approach is the most effective learning condition among those tested (as indicated by the Kruskal-Wallis test result of  $H(2) = 2.13$  and a p-value of 0.34), we can observe that the results show a positive trend in the desired direction.

### 3.1 Qualitative student feedback

In order to examine the potential impact of time constraints on learning, we solicited feedback from an additional group of students that learned under a two-minute time constraint. They were allowed to learn from both the video lecture and the summary within the given time frame. After the experiment, we asked the students to fill out a qualitative feedback survey about their study methods, specifically if they utilized the summary, video, or both.

Overall, we found that under timed conditions, students tended to use summaries over video lectures when both were available just as they would do when studying before an exam when time constraints make summaries more useful. We report three representative quotes from the student responses:

*"Summaries are helpful to get an overview of lecture."*

*"Used mainly the timestamps and the summary, didn't pay too much attention to the video itself."*

*"I initially did not look at the lecture summarizer because the material was new to me and as a result it seemed better to take in a larger quantity of material with new details. However, over time, as I began to get confused or did not recall all details about the lecture I started looking at the summaries. This is where I felt the lecture summaries were particularly valuable - to reinforce details about the lecture that I might have overlooked. Initially, without the context of having already watched the lecture, the summaries were not useful, but with this context existing as a sort of partial lattice structure in my head, the summaries became useful for filling the gaps that were missing from that structure."*

## 4 Conclusion

Our work shows that students reviewing both the lecture video and the automatically generated summary have a performance improvement from pre-test to post-quiz. This suggests that accompanying lecture videos with automatically generated summaries does improve the studying experience. As online learning becomes more ubiquitous, incorporating automatically generated summaries with videos can enhance students' overall learning experience.

## 5 Limitations

Our user study tests students on only two short lecture videos which are pre-recorded and carefully edited. Future work should test the efficacy of the summaries under a wider range of conditions: pre-recorded videos versus live lectures, lectures and summaries of different lengths, and a wider range of topics and disciplines.

Overall, our experiments compare three different conditions. Adding other conditions might have shed light on the relative value of automatic summaries. For instance, if we limit the time available for participants to prepare before taking the quiz, and at the same time track the amount of time spent on summaries and/or videos, then that could give better insights into how students would utilize the two sources differently with limited time constraints. Finally, we could also contrast the usefulness of summaries versus transcripts.

## Acknowledgements

This research is based upon work supported in part by the DARPA KAIROS Program (contract FA8750-19-2-1004), the DARPA LwLL Program (contract FA8750-19-2-0201), the IARPA HIATUS Program (contract 2022-22072200005), and the NSF (Award 1928631). Approved for Public Release, Distribution Unlimited. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA, IARPA, NSF, or the U.S. Government.

We would also like to thank Bryan Li and Rotem Dror for their helpful suggestions.

Finally, we extend our sincere appreciation to Professor Andrew Head for his guidance in experiment design and insightful discussions.

## References

- Stephen Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-david, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Fries, Maged Alshaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Dragomir Radev, Mike Tian-jian Jiang, and Alexander Rush. 2022. [Prompt-Source: An integrated development environment and repository for natural language prompts](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 93–104, Dublin, Ireland. Association for Computational Linguistics.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News summarization and evaluation in the era of GPT-3. *arXiv preprint arXiv:2209.12356*.
- Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.
- Juho Kim, Philip J Guo, Carrie J Cai, Shang-Wen Li, Krzysztof Z Gajos, and Robert C Miller. 2014. Data-driven interaction techniques for improving navigation of educational videos. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 563–572.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.
- Swaroop Mishra, Daniel Khoshabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. [Cross-task generalization via natural language crowdsourcing instructions](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.
- Amy Pavel, Colorado Reed, Björn Hartmann, and Maneesh Agrawala. 2014. Video digests: a browsable, skimmable format for informational lecture videos. In *UIST*, volume 10, pages 2642918–2647400. Cite-seer.
- Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Atsushi Shimada, Fumiya Okubo, Chengjiu Yin, and Hiroaki Ogata. 2017. Automatic summarization of lecture slides for enhanced student preview: Technical report and user study. *IEEE Transactions on Learning Technologies*, 11(2):165–178.

David C.D. van Alten, Chris Phielix, Jeroen Janssen, and Liesbeth Kester. 2020. Self-regulated learning support in flipped learning videos enhances learning outcomes. *Computers Education*, 158:104000.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Saelyne Yang, Jisu Yim, Juho Kim, and Hujung Valentina Shin. 2022. Catchlive: Real-time summarization of live streams with stream content and interaction data. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–20.

## A Model Details

We use the ‘text-davinci-002’ version of GPT-3 for our zero-shot and few-shot experiments, and as the basis for our fine-tuned davinci model. We use these parameters:

- Temperature: .6
- Frequency penalty: .7
- Presence penalty: .7
- Max tokens: maximum possible

To compute the maximum possible tokens for each API call we made to the model, we start with the total number of tokens that the model can process (4000 tokens for ‘text-davinci-002’, 2048 for our fine-tuned model) minus the number of tokens in the current prompt. We used OpenAI’s ‘GPT2TokenizerFast’ (from huggingface-transformers) to count tokens.

## B Example Summaries

Table 2 gives example summaries from 13 consecutive transcript chunks from a lecture on Neural Network Language Models given in an Artificial Intelligence course. This output is produced by our few-shot model. In the few-shot model, there are many repetitive outputs with several of the summaries beginning at *The speaker is discussing*.

Table 3 gives example summaries from 15 consecutive transcript chunks from a lecture on Vector-Space Semantic Models given in an Artificial Intelligence course. This table shows outputs from our

fine-tuned model. We can observe that the repetitions in the summary are gone and the style of the summaries has improved.

## C User Interface

Figure 2 shows the interface template for the video-and-summary condition group in the user study, where we have instructions and a lecture video on the left side, and a summary with timestamps on the right side followed by a quiz link. Students can scroll down on the lecture summary.

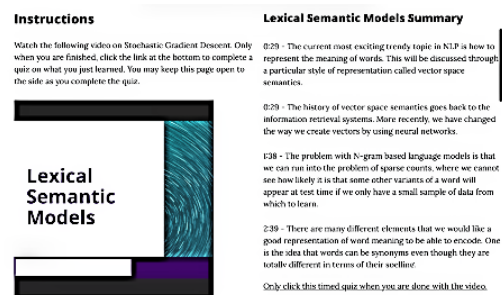


Figure 2: Video-and-Summary Interface

## D Pre-Quiz and Post-Quiz questions

### Lexical Semantic Pre-Quiz Questions

1. In N-gram language models, words are represented as
  - (a) Vectors
  - (b) Tokens
  - (c) Scalars
  - (d) Lists of ASCII codes
2. In the N-gram model, are the following words (introduce, introducing, introduced) treated as the same token? (T/F)
3. What is the LEAST important factor that we should consider while encoding word meanings?
  - (a) Words have synonyms
  - (b) Words that have opposite meanings
  - (c) Words that reflect different semantics roles
  - (d) The number of word occurrences
4. Semantic representation will not involve encompassing syntactic positions of the words. (T/F)

Raw Transcript Chunks	Automatic Summaries
<p>I feel a little bit like I'm landing an airplane coming in for the end of the semester filled with hundreds of people. But other than that. All right. How you doing? Thank you. Gave me a linchpin for the here. I appreciate it. Thank you. Yes. Yeah. It's a to cope with the mask of the students. Good. Thank you. Thank you. You too. Right. Yeah. All right. So now we're on to the very last topic of the class. My personal favorite natural language processing.</p>	<p>The speaker is discussing their favorite topic, natural language processing. They mention that they feel like they are landing an airplane when coming in for the end of the semester. The speaker thanks the person they are talking to and mentions that it is difficult for students to cope with mask wearing.</p>
<p>So uh we call it natural language processing to distinguish it from programming languages, natural language. Um and there's lots and lots of different elements of this field. So I'm gonna walk you through one particular topic. So we'll talk about um how we've been, how we've adapted to use neural networks to do the problem of language modeling. And then we'll talk more generally about how we can represent the meaning of words. So how do we come up with a computer interpret herbal representation of word meaning?</p>	<p>The speaker is discussing natural language processing and how it is different from programming languages. The speaker talks about how neural networks can be used to do the problem of language modeling. The speaker also talks about how computer representations of word meaning can be created.</p>
<p>Embedding and stores this in a matrix ah where we have the words in the vocabulary and each Word embedding of length M. So that's that 10, 30, 60 kind of length. Um And in this old paper they initialized this with singular value decomposition. But detailed doesn't really matter that much anymore. And the neural architecture that they use, I was just feed forward neural net. Okay, but here's my cartoon of what their neural network looks like. We've got our input words here. So we've got some number of input words, we want to predict the next word.</p>	<p>The neural network they use is a feed forward neural net. The input words are fed into the network, and the goal is to predict the next word.</p>
<p>And then we have a bunch of hidden layers and then we aggregate all the weights together into a weight matrix for each hidden layer and we have an output layer that we usually push through a softmax function. That changes the uh numbers here that are real valued numbers into a probability distribution. And that output layer, the number of nodes in the output layer can correspond to the number of classes that we have in our prediction problem. Okay. And then the math underlying, it is not that much different than we had for the logistic progression classifier.</p>	<p>Feed forward neural nets work. The math underlying the neural net is similar to the logistic progression classifier.</p>
<p>So the main difference is instead of taking the dot product between a weight vector and the input vector, we now take a matrix multiplication between the input vector X. And the weight matrix for that entire hidden layer. Right? But we still have the bias term. We still pump it through an activation function which could either be the sigmoid just like in logistic regression or it could be one of another uh set of things like rectified linear units or tangent. All right. And so that that then gives us the output of pin layer.</p>	<p>The main difference between a feed forward neural net and logistic regression is that with a feed forward neural net, we take a matrix multiplication between the input vector and the weight matrix. We still have the bias term and we still pump it through an activation function.</p>
<p>And then each hidden layer could act as the input to the subsequent hidden layer or to the final air. So each of these transmits uh its own output to the next player, just like the input to this first layer. And so each layer has its own set of weights. So these are the parameters of the model. And finally we put things through the softmax operation. Right.</p>	<p>Each hidden layer in a feed forward neural net transmits its output to the next layer. The weights for each layer are the parameters of the model. Finally, the output is put through a softmax operation.</p>
<p>And so last time we talked about training the networks Arianna, we use a cross entropy loss function, just like with logistic regression, we compute the gradients but we use uh the computation graphs and the back propagation algorithm. Okay, so how can we use these neural networks to do the problem of language modeling? So you'll recall from your language modeling homework that one of the goals of language modeling is to estimate the probability of some sequence of words that form a sentence.</p>	<p>Neural networks can be used for language modeling. They talk about how the goal of language modeling is to estimate the probability of a sentence.</p>
<p>So if we've got a sequence of words uh where we have them index from word one up towards the end been we can use the chain rule for probability to say the probability of each next word, depends on all the previous words. And the simplification that we've made in our language modeling homework, in our n gram based language modeling homework was to make a Markov assumption where we said the probability of the next word only depends on a short sequence of words in its history.</p>	<p>The probability of each next word in a sentence depends on all the previous words. The simplification that was made in the language modeling homework was to make a Markov assumption where the probability of the next word only depends on a short sequence of words in its history.</p>
<p>So it doesn't depends on the whole world history, just a short sequence. And the advantage that that had, was it allowed us a straightforward mechanism for estimating the probabilities. Alright, so once we make that Markov assumption, estimating the probabilities of each next word, given that limited history is really just a matter of going through and counting up that sequence of those and words and or and divided by the total number of words. Words with that preceding history plus any other words. Right. And so that would just be the maximum likelihood estimate.</p>	<p>The Markov assumption allows for a straightforward mechanism of estimating probabilities. Estimating the probability of each next word is just a matter of counting up the sequence of words and dividing by the total number of words with that preceding history.</p>

Table 2: Example summaries generated by our few-shot summarizer

Raw Transcript Chunks	Automatic Summaries
Any questions about the homework assignments, um, or the reading from, for the Ngrams? Nope. Okay. Um, so as friendly reminders, homeworks do at midnight, you have a total of five free late days for the course. Uh, you could use them at any time without asking the maximum number of late days you could use on any assignment is two. So, um, but I also recommend that maybe you don't. Uh, use them early if at all possible, because then you're just stuck in this thing with being a couple days behind on the next homework.	There are five free late days for the course, which can be used at any time. However, it is recommended that students do not use them early if at all possible.
So they're a blessing and a curse. Um, the next homework will be out shortly. I'm meeting up with one of the TAs to look over the revisions of it from last year, this afternoon. So with any luck, we'll get up, get it up this afternoon, even before, uh, homework number three is due. So this homework will be, uh, implementing the style of, uh, vector space models that we talked about, um, on Monday, which are the, uh, uh, long sparse vectors that you get via accounting. And then interesting transformations on the counts.	The next homework will be implementing vector space models using the counts from a corpus.
Like the TF IDF, the term frequency, inverse document frequency one, and also the positive pointwise mutual information, which we didn't cover in class, but which is included in the textbook in chapter six. So. Please do read that section. Um, and, uh, we'll sh we'll like, have you, uh, analyze a Corpus of Shakespeare's writings? So you'll be able to say like, uh, for this term by document matrix, pull out the column vectors, representing the documents and compare, uh, pairs of Shakespeare's plays.	The term frequency, inverse document frequency, and positive pointwise mutual information are all types of vector space models. The class will analyze a corpus of Shakespeare's writings using these techniques.
So, um, when people in the English department are studying Shakespeare, they categorize, uh, his plays into like dramas, comedies and histories. So it might be interesting to see whether, uh, the plays that are in those conceptual categories established through literature, um, have a higher co-sign similarity with each other than with the other categories. So that's one potential analysis that you could do, um, and then read the textbook. Chapter six, that'll be the quiz that'll, uh, be released this week and will be due again on Monday at midnight. Okay.	One potential analysis is to see whether plays in the same conceptual category have a higher co-sign similarity with each other than with the other categories.
So, uh, uh, last time we were talking about these term by document matrices that we can construct through counting. Uh, we talked about one of the two transformations that we could do to those by applying term frequency, inverse document frequency. The other in the textbook is PPMI. Um, and we also talked about how you could move from that idea that was developed in information retrieval, which is really, uh, really conceptualized the value of those matrices as a way of retrieving similar documents.	The term by document matrix is a way of retrieving similar documents.
So if your query was thought of as a document, you could pull related queries or sorry, related documents to that query by querying the co-signed similarity for all the documents in your term by document matrix. Um, and then we saw how you could extend that term by document matrix idea to get word semantics by having a term by context matrix or a term by term matrix. Um, so those term by term matrixes are parameterized in a lot of different ways. We could think about how many words of context we want to take into account.	We can query a term by document matrix for related documents by querying the co-signed similarity for all the documents in the matrix. We could also think about how many words of context to take into account when computing semantic similarity.
We could think about, um, adding interesting linguistic context. Like we saw through the dirt method where they had dependency information. So instead of immediately adjacent words, they looked at, uh, parent of child of grandparent, of grandchild of et cetera, but they all kind of had the property that the vectors that resulted from these various methods for, uh, creating the term by context matrix meant that the representation of words were long and sparse the length of the vectors tended to be some function of the vocabulary size.	We could think about adding interesting linguistic context to our word representations by looking at words that are higher up in the dependency tree. The length of the vectors tended to be some function of the vocabulary size.
Um, and the sparsity results in effect that by looking at a, a sliding window around a word, you're not gonna encounter that word. Co-occurring with all words in the vocabulary of English. So there's gonna be lots of zeros. So the, the place that we pivoted to at the end of last lecture was to start looking at the more modern, um, representation that we use for words still in the same vector space idea, called word embeddings, where the major difference is the vectors themselves.	The modern representation for words is called word embeddings. The major difference is that the vectors themselves are learned through training.
Instead of being the length, the size of the vocabulary are gonna be much, much shorter. We're going to be able to specify however many dimensions we want to use to encode the, the, uh, representation of the word. And usually we pick some relatively small number of dimensions, like on the order of 100 or 300.	We can specify as many dimensions as we want for the representation of a word. Usually we pick some small number, like 100 or 300.
um, and partially as a result of picking a much smaller number and as a result of how we are then going to train the values, uh, to be included in the representation of each word, the vectors then move from being sparse vectors with lots of zeros to dense vectors, where we have almost no zeros. So the algorithm that we briefly looked at last time, um, uh, was called word Tove, which produces these dense vectors.	The algorithm that is used to generate word embeddings is called word2vec. The vectors that are produced are dense vectors, which is different from the sparse vectors that we have seen before.
So the value of these dense vectors versus the sparse vectors, um, and the value of them be having a relative few number of dimensions is that they're much easier, uh, to use for things like machine learning. So for instance, if you were to train your classifier to say, is this word simple or complex, you could actually use those a hundred dimensions as features in your classifier.	The value of dense vectors for machine learning is that they are much easier to use than sparse vectors.

Table 3: Example summaries generated by our fine-tuned summarizer

### Lexical Semantic Post-Quiz Question

1. Vector space semantics is a representation of word meaning. (T/F)
2. In N-gram language models, words are represented as
  - (a) Vectors
  - (b) Tokens
  - (c) Scalars
  - (d) Lists of ASCII codes
3. In the N-gram model, the following words (introduce, introducing, introduced) are treated as the same token. (T/F)
4. What are the drawbacks of simply using N-gram models?
  - (a) We really didn't understand that there was a relationship between those different variants of the same underlying word
  - (b) We can run into the problem of sparse counts
  - (c) Both of A and B
  - (d) None of A and B
5. When we encode word meanings, we should consider the property that words can be synonyms, meaning that they have similar meanings to other words that are totally different. (T/F)
6. What is the LEAST important factor that we should consider while encoding word meanings?
  - (a) Words have synonyms
  - (b) Words that have opposite meanings
  - (c) Words that reflect different semantic roles
  - (d) The number of word occurrences
7. Semantic representation will not involve encompassing syntactic positions of the words. (T/F)
8. WordNet is an example of \_\_\_ knowledge base.
  - (a) Syntactical
  - (b) Lexical
  - (c) Grammatical
  - (d) Pronunciation
9. WordNet does not encode hierarchical organization of words. (T/F)
10. Which of the following refers to words that are more general than the current word?
  - (a) Hypernym
  - (b) Hyponym
  - (c) Synonym
  - (d) Antonym

### Stochastic Gradient Descent Pre-Quiz Questions

1. At each step, gradient descent finds out the direction along which the function changes the most quickly, and moves in this direction. (T/F)
2. In gradient descent, at each step, what should we know to update the weight?
  - (a) Previous weight, learning rate, slope value
  - (b) Previous weight, learning rate, previous function value
  - (c) Previous weight, previous function value
  - (d) Learning rate, slope value
3. What is the role of learning rate in gradient descent?
  - (a) To decrease the weights and avoid very large weights
  - (b) To control the step size of our move in gradient descent at each step
  - (c) To learn from the training set
  - (d) To account for overfitting
4. Gradient descent can not be used for weights with multiple features. (T/F)

### Stochastic Gradient Descent Post-Quiz Questions

1. At each step, gradient descent finds out the direction along which the function changes the most quickly, and moves in this direction. (T/F)
2. In gradient descent, at each step, what should we know to update the weight?
  - (a) Previous weight, learning rate, slope value

- (b) Previous weight, learning rate, previous function value
  - (c) Previous weight, previous function value
  - (d) Learning rate, slope value
3. What is the role of learning rate in gradient descent?
- (a) To decrease the weights and avoid very large weights
  - (b) To control the step size of our move in gradient descent at each step
  - (c) To learn from the training set
  - (d) To account for overfitting
4. What might be the problem when our learning rate is too big for convex functions?
- (a) We will have very large weights at the end
  - (b) We will have very small weights at the end
  - (c) We will move back and forth in gradient descent update and never find the global minimum
  - (d) There is no problem with a very large learning rate
5. For logistic regression, gradient descent can always find the global minimum of its loss function. (T/F)
6. Gradient descent can not be used for weights with multiple features. (T/F)
7. For convex functions, gradient descent with a reasonable learning rate can always find the global minimum. (T/F)
8. For convex functions, the starting point where we start gradient descent is not important. (T/F)
9. Gradient descent is a method that uses which of the following to determine the minimum of a function
- (a) The function's current value
  - (b) The function's intercept
  - (c) The function's slope
  - (d) The function's maximum value
10. Gradient Descent is guaranteed to find the minimum of the logistic regression loss function because

- (a) We use very powerful machines to run the method
- (b) The loss function is convex
- (c) The loss function is concave
- (d) We start gradient descent from a carefully chosen point

## E Summaries From Our User Study

### E.1 Lexical Semantic Lecture Summary

0:29 - The current most exciting trendy topic in NLP is how to represent the meaning of words. This will be discussed through a particular style of representation called vector space semantics.

0:29 - The history of vector space semantics goes back to the information retrieval systems. More recently, we have changed the way we create vectors by using neural networks.

0:29 - The problem with traditional language models is that they do not understand the relationship between words. In this class, we will discuss ways to create neural language models that do have this understanding.

1:38 - The problem with N-gram based language models is that we can run into the problem of sparse counts, where we cannot see how likely it is that some other variants of a word will appear at test time if we only have a small sample of data from which to learn.

2:39 - There are many different elements that we would like a good representation of word meaning to be able to encode. One is the idea that words can be synonyms even though they are totally different in terms of their spelling.

2:51 - We will discuss how to measure similarity between words, as well as how to understand the opposite and similar meanings of words.

3:36 - There can be words that reflect different semantic roles, and words that have a positive or negative connotation.

3:25 - Entailment is an important aspect of word meaning.

6:48 - Entailment can be mapped onto language in a way that reflects the meaning of words.

8:47 - Entailment can be used to make inferences. For example, if we know that all animals have an old and their artery, then we can infer that dogs must have an old and their artery.

4:51 - The ability to use logic as a representation of the meaning of language would give us a very powerful machinery for handling inferences and entailments.

8:46 - The downside of using formal logic as a representation for the meaning of language is that we have to acquire that knowledge. There are, however, resources that have been created that help with this problem. One very important resource is called WordNet, which is a lexical knowledge base containing the meaning of words.

7:23 - WordNet has synonym sets that represent different senses of a word. Each sense of the word is represented as a distinct concept.

8:59 - The WordNet resource encodes the meaning of words in a way that reflects the hierarchical organization of words in the language.

8:47 - In WordNet, a dog is a kind of canine and a domesticated animal. Clicking on each of these concepts shows how they are related through inheritance.

8:46 - In order to make an entailment, we need to be able to walk through the different levels of hierarchy in WordNet.

## **E.2 Stochastic Gradient Descent Lecture Summary**

0:00 - We want to find a parameter setting that minimizes this loss over all of the items in our training set. Theta is the set of parameters that we have at our model.

0:54 - We want a good setting of theta that minimizes the average loss across our training set using the cross entropy loss function.

1:52 - The algorithm that is used to find the optimal parameter setting is called gradient descent.

1:36 - The algorithm for finding the optimal parameter setting is called gradient descent. The algorithm uses a method for pushing around values in a weight vector to find the optimal setting.

2:23 - The algorithm uses a method for pushing around values in a weight vector to find the optimal setting. The analogy for thinking about this is you're in a canyon and you want to find your way down to the river.

2:51 - The idea of a function and what its minimum point is can be used to understand the idea of gradient descent.

3:20 - The loss function for logistic regression is convex, which means there's just one minimum for logistic regression. As a result, gradient descent starting from any point is guaranteed to find the minimum.

4:00 - The loss function for logistic regression looks like this. We can decrease  $w$  by pushing it in

this direction and we can increase  $w$  by pushing it in this direction.

4:22 - We want to find the point where the loss is the lowest by computing the slope of  $w$  with respect to our loss function.

2:00 - We will compute the slope of  $w$  with respect to our loss function and take one step in the direction of the slope.

6:53 - The learning rate is the amount by which we step in the direction of the slope.

1:36 - The weight at each time step is the current weight at the previous time step minus the learning rate, which is the step size. The slope is the derivative of the loss function with respect to the weight, and then we add back in the learning rate.

7:22 - The reason the curve goes up after we cross the minimum is because this is just how we drew this particular loss function. The minimum is always going to be with respect to the loss.

9:20 - The minimum point is the best weight associated with one of our features.

8:30 - This is a convex optimization problem, which means there's only one minimum. Other types of problems are nonconvex, which means there can be multiple minimums.

9:90 - The idea of taking a step in the direction of the slope may not work for nonconvex problems, which are problems with multiple minimums.

9:43 - The learning rate is how much we should step in the direction of the slope. There's interesting literature on how to set the learning rate for nonconvex problems.

10:43 - We want to use the intuition of moving left and right for a single value to move left and right for multiple variables.

11:14 - We will take the gradient of the weight across many dimensions and use that to find the minimum.

11:14 - We will use the intuition of moving left and right for a single value to move left and right for multiple variables.

11:25 - The intuition is that moving left and right for a single value should move left and right for multiple variables.

## **F Student performance on each quiz**

From the pre-quiz and post-quiz results, as shown in Tables 4 and 5, we can calculate the increase in percentages of mean correctness of the different conditions for Lexical Semantic lecture. Refer to



Table 6 for this analysis. Similarly, we calculate the increase for the Gradient Descent lecture, as shown in Table 7. Note that values are normalized when calculating the percentage increase.

Condition	Mean Correctness	Std Dev
Video	2.8	0.94
Summary	3.24	1.03
V+S	2.88	0.93

Table 4: Pre-Quiz results  
Lexical Semantic

Condition	Mean Correctness	Std Dev
Video	7.92	1.62
Summary	7.72	1.37
V+S	8.33	1.25

Table 5: Post-Quiz results  
Lexical Semantic

Condition	% increase
Video	13.14%
Summary	-4.69%
V+S	15.69%

Table 6: Percentage increase from Pre-Quiz to Post-Quiz results in Lexical Semantic Lecture

Condition	% increase
Video	26.48%
Summary	21.75%
V+S	35.33%

Table 7: Percentage increase from Pre-Quiz to Post-Quiz results in Stochastic Gradient Descent Lecture

# Automated Evaluation of Written Discourse Coherence Using GPT-4

**Ben Naismith**

Duolingo

ben.naismith@duolingo.com

**Phoebe Mulcaire**

Duolingo

phoebe@duolingo.com

**Jill Burstein**

Duolingo

jill@duolingo.com

## Abstract

The popularization of large language models (LLMs) such as OpenAI’s GPT-3 and GPT-4 have led to numerous innovations in the field of AI in education. With respect to automated writing evaluation (AWE), LLMs have reduced challenges associated with assessing writing quality characteristics that are difficult to identify automatically, such as discourse coherence. In addition, LLMs can provide rationales for their evaluations (ratings) which increases score interpretability and transparency. This paper investigates one approach to producing ratings by training GPT-4 to assess discourse coherence in a manner consistent with expert human raters. The findings of the study suggest that GPT-4 has strong potential to produce discourse coherence ratings that are comparable to human ratings, accompanied by clear rationales. Furthermore, the GPT-4 ratings outperform traditional NLP coherence metrics with respect to agreement with human ratings. These results have implications for advancing AWE technology for learning and assessment.

## 1 Introduction

Recent advances in large language models (LLMs; [Brown et al., 2020](#)), and in particular OpenAI’s GPT-4 model ([Eloundo et al., 2023](#); [OpenAI, 2023](#)), have led to a paradigm shift with regard to what machines can generate, such as coherent writing. We are now witnessing the potential power and exponential growth of AI in education, though the impact of LLMs used for educational purposes is still largely unexplored. For instance, applications not intended for educational purposes, such as ChatGPT, are being used in educational contexts – everyone with access to the internet can now ask ChatGPT to complete writing tasks, from generating outlines and ideas, to summarizing documents, to essay writing. With these novel capabilities, we can see immediate advantages, such as leveraging GPT-4 for instructional purposes (e.g., automatic

item generation, see [Attali et al., 2022](#)), and disadvantages (e.g., increased plagiarism, see [Eliot, 2022](#)). In addition, we are learning about current potential shortcomings of LLMs (e.g., hallucinations or low-quality content generation) due to miscalibrated expectations of what LLMs can do or the pitfalls of non-optimized prompt engineering.

To further our understanding of one innovative application of AI in education, this paper presents an exploratory evaluation of LLMs for automated writing evaluation (AWE). Specifically, it is the first study to our knowledge to examine GPT-4’s ability to provide a rating (score) and rationale for one aspect of writing quality – discourse coherence quality – in test-taker written responses to an online, high-stakes writing assessment item. Discourse coherence is notoriously challenging to satisfactorily assess using AWE, and as such, there is great value in determining whether state-of-the-art AI can be used to improve upon prior options. We believe that the method described in the paper should be generalizable to similar datasets that are publicly available. However, caution in the use of GPT-4 ratings is warranted due to limited reproducibility, the possibility of bias, and limited insight into the underlying processes that determine the ratings.

## 2 Background

In the field of AI in education, AWE is one of the most widely researched and mature areas. AWE systems evaluate written text quality ([Shermis and Burstein, 2003, 2013](#); [Attali and Burstein, 2006](#)) and are widely used for high-stakes writing assessment and instruction. These systems are informed by theoretical writing subconstructs (i.e., factors contributing to writing quality) described in human scoring rubric criteria such as grammatical accuracy, lexical sophistication, relevance, and discourse coherence. These rubric criteria are developed and used by educational testing organizations for scoring purposes and are often informed by

education policy (e.g., [Common Core Standards, 2010](#) and [Council of Europe, 2020](#)). AWE systems typically provide a holistic score that indicates the overall quality of writing, given a set of rubric criteria. The performance of these scores (accuracy) is then reported through human-system agreement, a well-studied evaluation measure that is typically quite high on modern systems (e.g., [Bridgeman, 2013](#)).

In recent years, large language models (and earlier models pretrained on unlabeled text) have been leveraged to good effect in various ways to improve AWE performance through the use of “transformers”, a type of deep learning neural network. For example, [Lagakis and Demetriadis \(2021\)](#) found that the best AWE performance was achieved through a model incorporating linguistic features with the BERT language model ([Devlin et al., 2019](#)). More recently, [Mizumoto and Eguchi \(2023\)](#) explored the capabilities of GPT-3 to holistically rate test-taker essays in the TOEFL11 corpus ([Blanchard et al., 2013](#)). The researchers showed Human-GPT-3 agreement rates to be reasonable (exact agreement 54.33%, adjacent agreement 89.15%). The model’s performance was then further improved by combining GPT ratings and a range of lexical, syntactic, and cohesion features, resulting in substantial Quadratic Weighted Kappa (QWK) of 0.61. Methodologically, it is important note that in their study, the same prompt was used in all conditions, and this prompt did not include examples or ask for rationales for the ratings. To our knowledge, there have been no similar studies with the newer GPT-4 or with comparing different prompt configurations to elicit ratings.

While AWE systems show strong performance for holistic scoring, scores for discourse coherence quality alone have been a challenging area of NLP research ([Hearst, 1997](#); [Barzilay and Lapata, 2008](#); [Burstein et al., 2013](#); [Somasundaran et al., 2014](#); [Lai and Tetreault, 2018](#)). Although some discourse features can be considered “surface-based,” for example, pronoun referents and transition terms used in a text, operationalizing aspects of coherence such as the relationship between ideas is less straightforward and involves labor-intensive annotations or less easily interpretable LLM-derived features. In particular, it may be difficult to tell whether LLM-generated “analyses” of a text actually reflect the same aspects of writing that superficially similar human-written analyses describe.

Further complicating coherence assessment is the fact that different disciplines, from linguistics ([Halliday and Hasan, 1976](#)) to cognitive psychology ([Graesser et al., 2004](#)), to education research ([Van den Broek et al., 2009](#)), share slightly different views about how coherence is constructed by readers of a text. However, a common thread is that discourse coherence pertains to the textual continuity or flow of a text, that is, the overall sense of unity and meaning that is conveyed by a text. Within the construct of discourse coherence, assessment rubrics often directly or indirectly refer to subconstructs such as clarity (how easy to understand ideas and purpose; readability; and impact of lexis/grammar on coherence); flow (sequence/progression of ideas; use of linking words; and referencing); structure (appropriacy of paragraphing; introducing/concluding; and connection between topics); and effect on reader (naturalness of cohesion; appropriacy of cohesive features; repetitiveness; and helpfulness to reader for understanding the response).

### 3 Methods

In this section we describe the dataset of test-taker responses and the processes for evaluating them through human and automated means.

#### 3.1 DET coherence (DET-Coh) dataset

The DET coherence (DET-Coh) dataset contains test-taker written responses from the operational Duolingo English Test (DET). The DET is a high-stakes English language test whose primary use is for higher-education admissions. One of the writing tasks, *Writing Sample*, is an independent writing task in which test takers respond to a prompt requiring them to produce a persuasive or narrative extended piece of writing in five minutes (see [Cardwell et al., 2023](#), for further details). *Writing Sample* is scored using AWE; the scoring model includes features to assess the writing subconstructs of Content, Discourse coherence, Grammar, and Vocabulary.

In total, there are 500 written responses in the DET-Coh dataset, sampled from the operational DET during a 7-month span in 2022. DET-Coh was deliberately constructed and stratified so that it contains an equal distribution of males and females, as well as an equal distribution of the seven most common first-language groups in the DET test-taker population (Chinese, Arabic, Spanish, Telugu, En-

glish, Bengali, Gujarati). An approximately even distribution of proficiency levels was also ensured based on DET automated scoring models. These levels align with the levels of the Common European Framework of Reference (CEFR; Council of Europe, 2001, 2020), an international standard for describing language ability, ranging from level A1 (basic) to C2 (proficient) on a six-point ordinal scale.

### 3.2 Human scoring

Test-taker writing responses were scored by four expert raters, each with second language (L2) teaching qualifications, extensive L2 teaching experience, and L2 assessment experience with international proficiency exams. Of the original 500 responses, 20 were double rated collaboratively for standardization, and 80 were rated independently by pairs of raters to assess interrater agreement. The interrater agreement for these 80 items was 0.72 exact agreement and 0.93 QWK, indicating excellent agreement. Having established rater reliability, the remaining 420 responses were rated by a single rater each.

All ratings were based on writing coherence task rubrics created for this study (see Appendix A, Table 2, for full rubric text). The rubric was developed using a 6-point, holistic scale that was based on the six levels/descriptors from the CEFR, other coherence research studies, and publicly-available rubrics from testing organizations. A rating of 0 was also given to blank or bad-faith responses in which the test taker did not attempt to respond to the prompt. In addition, one rater produced paragraph-long rationales for 12 of the ratings (two at each scale point) for the purposes of few-shot prompting (6 responses) and qualitative analysis (6 responses).

### 3.3 GPT-4 ratings and rationales

To elicit GPT-4 coherence ratings and rationales, we used the OpenAI Python API. The full prompt given to GPT-4 for each student response consisted of the following ordered elements:

- Task – a short paragraph explaining the task of rating the coherence of a written text written by a language learner in response to a prompt
- Rubrics – see Section 3.2 for description
- Guidelines – bullet point guidelines relating to expected terminology and style

- Examples – six training items removed from the dataset (one from each scale point), accompanied by expert ratings and/or rationales (depending on the condition) for the ratings based on the rubrics
- Prompt – the prompt the test taker responded to
- Response – the test taker’s response

Based on these elements, GPT-4 was called to complete three different conditions: 1) rating then rationale (rating-first), 2) rationale then rating (rationale-first), and 3) rating only (rating-only).

### 3.4 NLP coherence metrics

As a baseline, coherence ratings were predicted using a set of simple NLP features based on Coh-Metrix (Graesser et al., 2004):

- Binary overlap between sentence pairs: overlap of arguments, nouns, or word stems between two sentences
- Proportional overlap between sentence pairs: overlap of content words as a proportion of all content words in a sentence pair
- Coreference overlap: number of coreferent mentions between two sentences found using a neural coreference model (Lee et al., 2018)
- LSA similarity: measure of the similarity between two sentences calculated using an LSA model trained on a large sample of writing responses

Two versions of each feature were computed, one considering only adjacent sentence pairs (“local”), and one considering all pairs of sentences in a response (“global”). For each response, we fit a linear regression model using the features and human ratings for all other responses, then predicted the rating for the held-out response.

## 4 Results

### 4.1 Rating comparison

Ratings from GPT-4 and the baseline model are compared to the human ratings on all items not included in the prompt (Table 1); for double-rated items the second rating was used. The findings show that the baseline linear regression model is moderately predictive of the human ratings, reaching an adjacent agreement score of 0.82 and Spearman correlation ( $\rho$ ) of 0.47 despite its simplicity.

Metric	Human-baseline model	Human-GPT-4 (rating-rationale)	Human-GPT-4 (rationale-rating)	Human-GPT-4 (rating-only)
Exact agreement	0.36 (0.31-0.40)	<b>0.56</b> (0.52-0.60)	0.53 (0.49-0.58)	0.51 (0.46-0.56)
Adjacent agreement	0.82 (0.78-0.85)	0.96 (0.95-0.98)	<b>0.97</b> (0.95-0.98)	0.95 (0.93-0.97)
Cohen’s Kappa	0.13 (0.08-0.18)	<b>0.43</b> (0.38-0.48)	0.40 (0.36-0.46)	0.36 (0.31-0.42)
Quadratic Weighted Kappa	0.39 (0.33-0.45)	0.81 (0.79-0.84)	<b>0.82</b> (0.79-0.85)	0.78 (0.75-0.82)
Spearman’s rho	0.47 (0.39-0.53)	<b>0.82</b> (0.79-0.85)	<b>0.82</b> (0.79-0.85)	0.79 (0.76-0.83)

Table 1: Coherence rating agreement rates, with bootstrapped 95% confidence intervals (percentile). Bold indicates the best performance for a metric. All GPT-4 conditions have significantly better agreement with human ratings than the baseline model across all metrics. The two GPT-4 conditions which produce a rationale have marginally (but not significantly) better agreement than the rating-only condition.

All GPT-4 conditions significantly outperform this baseline model, obtaining a correlation of 0.82 with the human rating in the rationale conditions.

Inspired by Mizumoto and Eguchi (2023), we also experimented with a linear regression model that includes the GPT-4 rating as an additional feature along with the baseline features, potentially combining the strengths of the two models. However, unlike that work, we found that the combined model performs almost identically to the GPT-4 ratings on their own and so do not analyze it further.

The rationale-first condition could be interpreted as a form of chain-of-thought (CoT) prompting (Wei et al., 2022) which has been shown to improve performance on reasoning tasks. That work also hypothesized that showing examples with the reasoning after the answer in the prompt could improve performance, by drawing attention to relevant aspects of the tasks, but found it performed similarly to the baseline and worse than CoT prompting. By contrast, we find that GPT-4’s agreement is slightly improved by the use of rationales, regardless of their position. However, there are no significant differences between the agreement rates of any of the GPT-4 configurations, with all versions showing overlapping confidence intervals. These findings suggest that there is not a CoT effect for this task.

We focus on the rating-first condition for error analysis. GPT-4’s ratings have less variance than human ratings (0.37 vs 0.42), especially producing fewer 1, 5, and 6 ratings (most samples rated 1 by

humans are rated 2 by GPT-4). This behavior is actually in-line with a well-documented tendency of human raters, the central tendency effect, in which raters avoid the extremes of rating scales (McNamara et al., 2019). One hypothesis to account for this pattern is that GPT-4 is imitating trends found in its pre-training data. When GPT-4’s ratings differ from human ratings (n=143), they are also slightly but significantly lower on average ( $\mu = 3.17$  for GPT-4 in the rating-first condition vs  $\mu = 3.41$  for the comparable human rating,  $p=0.04$  with Welch’s t-test). In the rating-first condition, GPT-4 mentions “spelling” in 43% of rationales where its rating differs from the human rating, versus only 30% of equally rated rationales. Speculatively, this may indicate an oversensitivity to spelling errors; human raters may be better able to discern the intended word while GPT-4’s token-based representation may prevent such recognition.

## 4.2 Rationale comparison

The six human-generated rationales were compared to GPT-4 rationales in terms of their content and style. Figure 1 provides an example of a response with a 3 rating (CEFR B1; human and GPT-4 rating in agreement), answering a prompt about the advantages and disadvantages of using books, movies, and TV shows to learn about different cultures. Figure 2 shows the accompanying human and GPT-4 rationales. Of note, the trends exemplified in this set of examples hold true for all six pairs of human-

I think that we can learn different cultures from using books and watching movies and tv shows. We can be able to get to know how they live, eat, dance and party. There are many advantages and disadvantages learning about different cultures through books or movies. Firstly, the advantages of learning about the cultures are we can get to learn morals, codes and ethics of that certain culture and know their language, hardwork that they do in that characters and mostly the people mentality. Secondly, disadvantages are we acn also get encounter dru

Figure 1: Example test-taker response (3 rating / CEFR B1).

Human rationale	GPT-4 rationale
<p>The response has variable coherence. A relevant position is presented and a point in favor of using media to learn about different cultures is given. However, the topic is not developed, with the second point about disadvantages ending abruptly so that the intended message is not conveyed. There is some appropriate use of basic discourse markers (e.g., “Firstly”, “Secondly”). The language is sufficient to convey basic ideas on the topic, but lexis from the prompt is used repetitively and at times coherence is affected (e.g., “hardwork that they do in that characters”). Overall, the response produces a satisfactory effect on the reader, negatively impacted by language inaccuracies and lack of development.</p>	<p>The response has variable coherence. The writer begins by stating that it is possible to learn about different cultures through books, movies, and TV shows, and provides some examples of what can be learned. The writer then moves on to discuss the advantages and disadvantages of this approach. However, the response is unfinished and the disadvantages are not fully developed. The response demonstrates some organization, with the use of discourse markers such as “Firstly” and “Secondly” to link ideas. However, there are some inaccuracies in grammar and vocabulary that impact coherence, for example “we acn also get encounter dru”. Overall, the response is somewhat effective in conveying the intended message, but the unfinished nature of the response and inaccuracies in language limit the rating to a 3.</p>

Figure 2: Comparison of rationales for assessment of Figure 1.

GPT4 rationales we analyzed.

Comparing the content of the two rationales, there is a great deal of consistency, with both addressing the clarity, flow, structure, and effect on the reader. For example, both rationales describe how the writer’s position is initially presented and provide a specific example. The two rationales also note the same main weakness relating to the lack of development of the second point. The two rationales then move on to describe how discourse markers are used to achieve local coherence, even highlighting the same two examples of *Firstly* and *Secondly*. Examples of coherence negatively affected by language inaccuracies are then given, though different examples are used to exemplify this point in the two rationales. Finally, both rationales summarize the reason for the overall satisfactory effect on the reader.

Likewise, in terms of style, the GPT-4 rationale has clearly adopted the examples and followed the guidelines from the prompt. The rationales use

terminology such as *the writer* (rather than *the author/student/learner*), are written in the 3rd person, and are within the desired length range. The overall format of the rationale is also consistent, starting with an overall statement of coherence, moving to discuss each of the coherence subconstructs in turn, then closing with an overall description of the effect on the reader.

To further illustrate how GPT-4 rationales discuss and incorporate key concepts from the rubrics, we conducted a simple corpus analysis of key words. First, a frequency list was compiled of the most common words (tokens) in the rationales. We restricted this list to content words (nouns, verbs, adjectives, and adverbs) and only counted the first occurrence of each word in each rationale. Of interest, we noted commonly used words related to discourse coherence including *ideas* (n=509), *developed* (n=406), *impact* (n=297), *inaccuracies* (n=278), and [discourse] *markers* (n=264). Figure 3 presents a concordance of the first ten oc-

. The reader is able to discern some relevant ideas	but the response is not well-organized or developed ideas
a result the reader struggles to identify any relevant ideas	. There is no evidence of discourse features such ideas
the response contains a number of incomplete or incoherent ideas	for example , the issue of scales to travel ideas
lacks an overall structure appropriate for the task and ideas	are not clearly presented or arranged . The discursal ideas
has minimal coherence . The writer expresses two basic ideas	: that video conferencing applications are easy to learn ideas
lacks an overall structure appropriate for the task and ideas	are not clearly presented or arranged . Grammar and ideas
coherence . It is possible to discern some relevant ideas	, such as the writer’s decision to date ideas
lacks an overall structure appropriate for the task with ideas	not clearly presented or arranged . As a result ideas
is minimally coherent with the writer expressing two basic ideas	: that taking notes with pen and paper takes ideas
coherence . It is possible to discern some relevant ideas	such as that travel can provide information and ideas

Figure 3: Uses of the key term *ideas* in the GPT-generated rationales with local context.

currences of the most frequent of these key words, *ideas*, to provide the context in which this term is being used. Here we see that *ideas* are described in a number of ways, for example, *relevant*, *appropriate*, *basic*, and *incoherent*, all of which are descriptors used in the rubrics. As importantly, these *ideas* are discussed in terms of how they are presented and arranged in the response, and specific examples of test-taker ideas are listed, that is, there is a focus on content and meaning, not just mechanical use of linguistic features.

## 5 Discussion and conclusions

This study examined the effectiveness of using GPT-4 for assessing written discourse coherence of test-taker responses on a high-stakes English proficiency test. We found that GPT-4 is able to rate the coherence of writing samples with a good degree of accuracy in terms of agreement with the gold-standard human ratings; regardless of the exact order of the prompt (rating-first or rationale-first), the exact agreement rates were  $>0.5$  and the QWK  $>0.8$ . Prompts eliciting rating-only performed slightly worse, though not significantly so. Importantly, all permutations of the GPT-4 prompt greatly outperformed a baseline NLP model composed of traditional coherence features. Human-GPT-4 agreement rates could likely be improved with further tailoring of the prompt; for example, based on the qualitative analysis, we might suggest additional guidelines to lower the weighting that GPT-4 assigns to spelling errors as it may be overvaluing their importance.

Studies such as this one have important implications for the field of AWE. There is often a tension between designing features that are easily interpretable but provide limited signal (e.g., the number of discourse markers) versus features which are less clearly aligned with human rubrics but which may provide more predictive power (e.g., perplexity of

the response under a language model). The promise of ratings based on GPT-4 is that they may bridge this gap by providing quantitative features which seemingly are based on aspects of language of importance to the language assessment community. In the future we therefore expect to see research in a similar vein which looks at further optimizing prompts to elicit ratings and clear, interpretable rationales, especially for subconstructs of writing which have historically been a challenge to measure through automated means. In using LLMs in this manner, we could reduce the “epistemic opacity” of AWE processes (Ferrara and Qunbar, 2022), that is, modern automated assessment could become less of a black box, thereby improving stakeholder confidence in the results. Nevertheless, although these results are encouraging, it is important to recognize that the interpretability promised by generated rationales is limited: GPT-4’s rationales may not accurately reflect the process used to assign the ratings. In particular, rationales may present rationalizations for decisions actually grounded in biasing features, as was found to be true of CoT explanations in Turpin et al. (2023). Rationales should therefore not be treated as offering insight into the *process of generating* ratings, even when they provide true and relevant information about the response.

The fact that rationales do not reflect a “thought process” by GPT does not, however, reduce their value in all contexts. As suggested in Mizumoto and Eguchi (2023), rationales can support language learning by providing instantaneous feedback. In the context of test takers of the DET, rationales such as the ones in this report are particularly useful because they are based on task- and construct-specific rubrics. For example, test takers completing a practice test would greatly benefit from feedback tailored to the writing subconstructs, such as discourse coherence, that will be assessed under

operational test conditions. GPT-4 could also then be further beneficially exploited by querying it to produce an improved version of the test taker's own response; in other words, a personalized model answer.

Figure 4 is an example of one such model answer, revising the response from Figure 1. The same prompt as before was used for generating this revision, with the following amendment:

*Now, write a revised version of the following response with improved coherence according to the rubric. Stick closely to the original in content, and do not rewrite too extensively; simply improve the organization and complete unfinished ideas.*

In this revision, we see that the test taker's ideas are maintained, for example, the benefits of learning about how other cultures eat and dance. In addition, the appropriate use of some discourse markers from the original are left intact. In contrast, key coherence weaknesses from the original are addressed, most notably the lack of development of disadvantages and the language inaccuracies which impacted clarity. There remains some repetitiveness in the revision of language from the task prompt, but this issue did not prevent the revised response from being independently rated a 5 (CEFR C1) by both GPT-4 and a human rater. As such, this revision would seem a reasonable goal for this particular test taker.

On a broader level, the focus of our study, including the importance of transparency, is in line with the larger field of educational AI application development where responsible AI is a key focus (ATP, 2021; Dignum, 2021; ITC-ATP, 2021; Burstein, 2023; Department for Science, Technology & Innovation, 2023). As novel ideas, applications, and research questions emerge around the use of LLMs for educational purposes, it is essential that research communities investigating the use and impact of AI for education build a research agenda. In light of the need to ensure responsible use of AI in education, researchers need to anticipate and pressure test possible uses of AI for education to ensure fairness.

## Acknowledgements

We thank the raters for their contribution to the DET-Coh dataset.

I believe that we can learn about different cultures by using books, watching movies, and TV shows. Through these mediums, we can gain insights into how people from various cultures live, eat, dance, and celebrate. There are both advantages and disadvantages to learning about different cultures through books, movies, and TV shows. Firstly, the advantages include being able to learn about the morals, codes, and ethics of a particular culture. Additionally, we can become familiar with their language and observe the hard work and mentality of the characters portrayed in these mediums. On the other hand, there are also disadvantages to this approach. For example, we may encounter negative stereotypes or misrepresentations of a culture, which can lead to misunderstandings and misconceptions. Furthermore, relying solely on books, movies, and TV shows may not provide a comprehensive understanding of a culture, as these mediums often focus on specific aspects or situations. In conclusion, while there are benefits to learning about different cultures through books, movies, and TV shows, it is important to be aware of the potential drawbacks and seek additional sources of information to gain a more accurate and well-rounded understanding of various cultures.

Figure 4: Response revised by GPT-4.

## References

- Yigal Attali and Jill Burstein. 2006. [Automated essay scoring with e-rater® v.2](#). *The Journal of Technology, Learning and Assessment*, 4(3).
- Yigal Attali, Andrew Runge, Geoff T. LaFlair, Kevin Yancey, Sarah Goodwin, Yena Park, and Alina A. Von Davier. 2022. [The interactive reading task: Transformer-based automatic item generation](#). *Frontiers in Artificial Intelligence*, 5.
- Regina Barzilay and Mirella Lapata. 2008. [Modeling local coherence: An entity-based approach](#). *Computational Linguistics*, 34(1):1–34.
- Daniel Blanchard, Joel R. Tetreault, Derrick Higgins, A. Cahill, and Martin Chodorow. 2013. [TOEFL11: A corpus of non-native english](#). *ETS Research Report Series*, 2013:15.
- Bren Bridgeman. 2013. *Human ratings and automated essay evaluation*, pages 243–254. Routledge.



- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, et al. 2020. [Language models are few-shot learners](#). In *Advances in neural information processing systems*, volume 33, pages 1877–1901.
- Jill Burstein. 2023. [Duolingo English Test Responsible AI Standards](#). Technical report, Duolingo.
- Jill Burstein, Joel Tetreault, and Martin Chodorow. 2013. [Holistic discourse coherence annotation for noisy essay writing](#). *Dialogue & Discourse*, 4(2):34–52.
- Ramsey Cardwell, Ben Naismith, Geoffrey T. LaFlair, and Steven Nydick. 2023. [Duolingo English Test: Technical Manual](#).
- Council of Europe. 2001. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment*. Council of Europe Publishing, Cambridge, UK.
- Council of Europe. 2020. *Common European Framework of Reference for Languages: Learning, teaching, assessment – Companion volume*. Council of Europe Publishing, Cambridge, UK.
- Department for Science, Technology & Innovation. 2023. [A pro-innovation approach to AI regulation](#). White paper, Crown.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of NAACL*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Virginia Dignum. 2021. [The role and challenges of education for responsible AI](#). *London Review of Education*, 19(1):1–11.
- Lance Eliot. 2022. [Enraged worries that generative AI ChatGPT spurs students to vastly cheat when writing essays, spawns spellbound attention for AI ethics and AI law](#). *Forbes*.
- Tyna Eloundo, Sam Manning, Pamela Mishkin, and Daniel Rock. 2023. [GPTs are GPTs: An early look at the labor market impact potential of large language models](#). *arXiv preprint arXiv:2303.10130*.
- Steve Ferrara and Saed Qunbar. 2022. [Validity arguments for AI-based automated scores: Essay scoring as an illustration](#). *Journal of Educational Measurement*.
- Arthur C. Graesser, Danielle S. McNamara, Max M. Louwerse, and Zhiqiang Cai. 2004. [Coh-matrix: Analysis of text on cohesion and language](#). *Behavior Research Methods, Instruments, & Computers*, 36(2):193–202.
- Michael A. K. Halliday and Ruqaiya Hasan. 1976. *Cohesion in English*. Longman, London, UK.
- Marti A. Hearst. 1997. [Text tiling: Segmenting text into multi-paragraph subtopic passages](#). In *Computational Linguistics*, volume 23, pages 33–64.
- International Test Commission and Association of Test Publishers (ITC-ATP). 2022. [Guidelines for technology-based assessment](#). Technical report, International Test Commission and Association of Test Publishers, Washington, D.C.
- Paraskevas Lagakis and Stavros Demetriadis. 2021. [Automated essay scoring: A review of the field](#). In *Proceedings of the 2021 International Conference on Computer, Information and Telecommunication Systems (CITS)*, pages 1–6. Institute of Electrical and Electronics Engineers.
- Alice Lai and Joel Tetreault. 2018. [Discourse coherence in the wild: a dataset, evaluation and methods](#). In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 214–223. Association for Computational Linguistics.
- Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. [Higher-order coreference resolution with coarse-to-fine inference](#). In *Proceedings of NAACL*, pages 687–692. Association for Computational Linguistics.
- Tim F. McNamara, Ute Knoch, and Jason Fan. 2019. *Fairness, Justice and Language Assessment*. Oxford University Press, Oxford, UK.
- Atsushi Mizumoto and Masaki Eguchi. 2023. [Exploring the potential of using an AI language model for automated essay scoring](#). *Educational Technology Research and Development*.
- National Governors Association Center for Best Practices and Council of Chief State School Officers. 2010. [Common Core State Standards](#).
- OpenAI. 2023. [GPT-4 technical report](#). *arXiv preprint arXiv:2303.08774*.
- Mark D. Shermis and Jill Burstein. 2003. *Automated essay scoring: A cross-disciplinary perspective*. Routledge, Mahwah, NJ.
- Mark D. Shermis and Jill Burstein. 2013. *Handbook of automated essay evaluation*. Routledge, New York, NY.
- Swapna Somasundaran, Jill Burstein, and Martin Chodorow. 2014. [Lexical chaining for measuring discourse coherence quality in test-taker essays](#). In *The 25th International Conference on Computational Linguistics (COLING)*, pages 23–29.
- The International Privacy Subcommittee of the ATP Security Committee (ATP). 2021. [Artificial intelligence and the testing industry: A primer](#). Association of Test Publishers.
- Miles Turpin, Julian Michael, Ethan Perez, and Samuel R Bowman. 2023. [Language models don’t always say what they think: Unfaithful explanations in chain-of-thought prompting](#). *arXiv preprint arXiv:2305.04388*.

Paul Van den Broek, Charles R. Fletcher, and Kirsten Ridsen. 2009. [Investigations of inferential processes in reading: A theoretical and methodological integration](#). *Discourse Processes*, 16:169–180.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems (NeurIPS)*.

## A Discourse coherence rubrics

Rating	Description
6 (C2)	<p>The response is highly coherent: (1) the ideas and purpose of the response are completely clear, and lexical/grammatical choices effectively enhance coherence; (2) the response is smoothly-flowing, with a clear sequence of ideas which are cohesively linked using a range of discoursal features (including cohesive devices and referencing); (3) the response is logically and appropriately structured for the task, with topics effectively developed and expertly connected.</p> <p><i>Do the discoursal features have an excellent effect on the reader, such that they are completely natural and do not attract any attention; they are appropriate for the text type; and they help the reader to understand the ideas in the response?</i></p>
5 (C1)	<p>The response is coherent: (1) the ideas and purpose of the response are clear, and lexical/grammatical choices rarely impact coherence in any way; (2) the response has a clear progression and ideas are linked using a range of discoursal features (including cohesive devices and referencing), though there may be some under-/over-use; (3) the response is well-structured for the task, with topics appropriately introduced, developed, and concluded.</p> <p><i>Do the discoursal features have a very good effect on the reader, such that they are mostly natural; they are appropriate for the text type; and they allow the reader to follow along easily?</i></p>
4 (B2)	<p>The response is mostly coherent: (1) the ideas and purpose of the response are clear, and lexical/grammatical choices generally do not impact coherence though they may lead to some instances of confusion; (2) the response has a generally clear overall progression and ideas are generally linked effectively despite some inaccurate or unnatural use of cohesive devices and referencing; (3) the response is generally well-structured for the task, with topics usually developed in some detail though some arguments may lack clarity.</p> <p><i>Do the discoursal features have a good effect on the reader, such that they are mostly appropriate despite some inaccuracies or repetitiveness, and they allow the reader to follow along?</i></p>
3 (B1)	<p>The response has variable coherence: (1) the reader can generally follow the overall purpose and the main points made by the writer, though lexical/grammatical choices impact coherence at times; (2) the response demonstrates some organization, linking discrete elements in a linear sequence, though the use of referencing and cohesive devices may be inaccurate and the overall progression may be unclear; (3) the response contains evidence of some structure appropriate for the task, though topics are not always developed, clearly distinct, or clearly connected, and argumentation may lack coherence.</p> <p><i>Do the discoursal features have a satisfactory effect on the reader, such that they are somewhat effective in conveying the intended message, despite inaccuracies or repetitiveness which impact coherence and cohesion?</i></p>
2 (A2)	<p>The response has minimal coherence: (1) it is possible to discern some relevant ideas, though the overall purpose of the response may be incoherent and the lexical/grammatical choices lead to breakdowns in coherence other than for basic ideas; (2) there is limited evidence of organizational features including cohesive devices and referencing, and when used, such features may be inaccurate and lead to breakdowns in coherence; (3) the response lacks an overall structure appropriate for the task and ideas are not clearly presented or arranged.</p> <p><i>Do the discoursal features have a poor effect on the reader, such that they are mostly not effective in conveying the intended message, with inaccuracies or repetitiveness often impacting coherence and cohesion?</i></p>
1 (A1)	<p>The response mostly lacks coherence: (1) it is a strain on the reader to identify points the writer is trying to make, with lexical/grammatical choices greatly impacting coherence throughout; (2) there is no apparent logical organization of ideas other than simple isolated phrases, with no or minimal/inaccurate use of discoursal features such as linking and referencing; (3) there is no overall structure appropriate for the task and ideas are difficult to discern.</p> <p><i>Do the discoursal features have a very poor effect on the reader, such that they are mostly not effective in conveying the intended message, with inaccuracies or repetitiveness often impacting coherence and cohesion?</i></p>
0	<p>There is no response or the test-taker is not responsive to the prompt in good faith, e.g., the test taker repeats the prompt but does not respond to it, or the the test taker intentionally goes off-task in some way to “trick” the system, for example, by writing random words, writing in a non-English language, writing random strings of letters, or giving a memorized/plagiarized off-topic response.</p>

Table 2: Discourse coherence rubrics used for human rating and GPT prompting

# ALEXSIS+: Improving Substitute Generation and Selection for Lexical Simplification with Information Retrieval

Kai North<sup>1</sup>, Alphaeus Dmonte<sup>1</sup>, Tharindu Ranasinghe<sup>2</sup>  
Matthew Shardlow<sup>3</sup>, Marcos Zampieri<sup>1</sup>

<sup>1</sup>George Mason University, USA, <sup>2</sup>Aston University, UK

<sup>3</sup>Manchester Metropolitan University, UK

knorth8@gmu.edu

## Abstract

Lexical simplification (LS) automatically replaces words that are deemed difficult to understand for a given target population with simpler alternatives, whilst preserving the meaning of the original sentence. The TSAR-2022 shared task on LS provided participants with a multilingual lexical simplification test set. It contained nearly 1,200 complex words in English, Portuguese, and Spanish and presented multiple candidate substitutions for each complex word. The competition did not make training data available; therefore, teams had to use either off-the-shelf pre-trained large language models (LLMs) or out-domain data to develop their LS systems. As such, participants were unable to fully explore the capabilities of LLMs by re-training and/or fine-tuning them on in-domain data. To address this important limitation, we present ALEXSIS+, a multilingual dataset in the aforementioned three languages, and ALEXSIS++, an English monolingual dataset that together contains more than 50,000 unique sentences retrieved from news corpora and annotated with cosine similarities to the original complex word and sentence. Using these additional contexts, we are able to generate new high-quality candidate substitutions that improve LS performance on the TSAR-2022 test set regardless of the language or model.

## 1 Introduction

Text simplification (TS) is utilized in educational technologies to automatically reduce the complexity of texts making them more accessible for various target populations, including children, second language learners, individuals with low-literacy, or those suffering from a reading disability, such as dyslexia or aphasia (Paetzold and Specia, 2017b; North et al., 2022c, 2023).

With an increase in online learning, there has emerged a greater need for personalized learning

platforms (McCarthy et al., 2022). These educational technology platforms need to be accessible to users. TS systems provide a solution by adapting content specifically for a user’s level of literacy in a given target language (Figure 1).

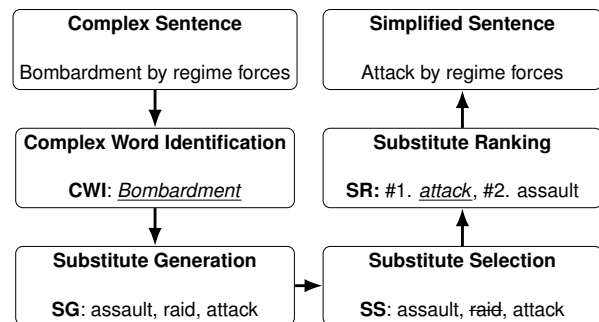


Figure 1: LS Pipeline. We only focus on SG and SS.

Lexical simplification (LS) is a precursor to TS (Paetzold and Specia, 2017b; North et al., 2022c). LS replaces challenging words, known as complex words, with simpler alternatives, hereby referred to as candidate substitutions. The generation of these candidate substitutions is known as substitute generation (SG) (Qiang et al., 2020; North et al., 2022b; Ferres and Saggion, 2022). SG attempts to predict viable candidate substitutions for an identified complex word. These candidate substitutions need to be easier to read and comprehend as well as be semantically similar to the identified complex word in its given context. An LS system would identify a complex word, for instance, “*bombardment*”, as being in need of simplification. It would then suggest such words as “*attack*”, “*assault*” or “*raid*” as being valid candidate substitutions since they are shorter, more familiar to a set of annotators, or are found to be more frequent within a reference corpus. These candidate substitutions would then be passed to a TS system that would, in turn, simplify any unnecessary syntax resulting in an easier to read sentence.

Various methods have been applied to the task

of SG for LS. The use of pre-trained LLMs trained with a masked language modeling (MLM) objective is the most favored approach to this task and has been shown to outperform other methods (Saggion et al., 2022). However, the performance of MLM for SG is largely dependent on the model and the dataset it has been pre-trained on (North et al., 2022a). This hinders SG for LS, since many LS datasets contain a small number of instances or a low number of gold candidate substitutions (North et al., 2022b). As such, participants in the TSAR-2022 shared-task on LS (Saggion et al., 2022) were forced to conduct zero-shot predictions for SG due to insufficient training data.

This paper presents ALEXSIS+ and ALEXSIS++<sup>1</sup>, two new datasets for LS. We propose an information retrieval (IR) approach that utilizes collected data from news sources. These two datasets contain 50,000 additional contexts for the original 1,500 complex words of the ALEXSIS dataset (Štajner et al., 2022), and can be used to generate accurate candidate substitutions for SG in a zero-shot condition identical to that at TSAR-2022 (Saggion et al., 2022). We demonstrate how these new datasets can be applied to any language or model without re-training or fine-tuning to increase LS performance on the TSAR-2022 test set. ALEXSIS+ and ALEXSIS++ were also constructed using only the data available to the participants of the TSAR-2022 shared-task, making our IR approach to SG, and later substitute selection (SS), highly adaptable. Furthermore, unlike ALEXSIS, which only features candidate substitutions, ALEXSIS+ and ALEXSIS++ feature multiple sentences per complex word providing new contexts that serve as useful data for MLM. Finally, as the approach doesn't require manually annotating data as in the original ALEXSIS, it can be used to improve the same unsupervised LS approaches purposed for the TSAR-2022 shared-task.

The main contributions of this paper are:

1. We propose an IR-based language independent approach to SG and SS. To the best of our knowledge, data collection efforts of this kind have not been explored within the context of LS.
2. We release ALEXSIS+ and ALEXSIS++, two new datasets for LS which open new avenues

<sup>1</sup>ALEXSIS+ and ALEXSIS++ have been made publicly available at: <https://github.com/LanguageTechnologyLab/ALEXSIS2.0>

for unsupervised models with performances surpassing those reported at TSAR-2022.

3. We evaluate multiple models on the two datasets, and we discuss the results in detail.

## 2 Related Work

**Pipeline** The LS pipeline contains three sub-tasks (Figure 1). The first of these is SG which produces  $k = n$  of candidate substitutions for a complex word with  $k$  normally being set to  $k = [1, 3, 5, \text{ or } 10]$  (Paetzold and Specia, 2017b). The top candidate ( $k@1$ ) is then chosen to replace the complex word. This candidate is selected through two additional sub-tasks: SS, and substitute ranking (SR). SS filters inappropriate candidate substitutions by removing candidates that are equal to or semantically dissimilar to the complex word along with those that are inappropriate in that context. SR orders a list of candidate substitutions based on their appropriateness. Techniques for SS and SR include sorting or filtering on frequency (North et al., 2022a), word length (Paetzold and Specia, 2017b), cosine similarity between word embeddings (Song et al., 2020). More recent approaches have used regression (Maddala and Xu, 2018), referred to as lexical complexity prediction (LCP) designed to replace binary CWI (North et al., 2022c), as well as prompt learning (Aumiller and Gertz, 2022). The TSAR-2022 shared-task (Saggion et al., 2022) challenged participating teams with generating a list of  $k = 10$  candidate substitutions for a given complex word in English, Spanish, and Portuguese. One of TSAR's key findings is that SR is less impactful on overall LS performance compared to SG, regardless of language. Systems that relied solely on SG with minimal SR outperformed those that employed various SR methods. LS systems that relied purely on SG were often found to have used a pre-trained LLM trained with an MLM objective to generate their top- $k$  candidate substitutions. We therefore focus on an IR approach that only improves the performance of LS through the generation and selection of additional candidate substitutions for the TSAR test set.

**Masked Language Modeling** MLM for LS involves feeding two concatenated sentences into an LLM separated by the [SEP] special token. The first sentence is the unaltered original sentence. The second sentence is the same as the original sentence, however, the target complex word is converted into the [MASK] special token. The LLM

	ALEXISIS			ALEXISIS+			ALEXISIS++
	EN	ES	PT	EN	ES	PT	EN
Languages							
Total unique complex words	386	381	386	386	381	386	386
Total unique contexts	386	381	386	12,831	13,353	13,541	33,149
<b>Total unique candidate subs.</b>	3,676	3,775	3,404	120,645	101,470	99,563	289,379
Avg. # of unique contexts per complex word.	1	1	1	54.60	95.90	60.15	108.18

Table 1: Comparison of the ALEXISIS, ALEXISIS+, and ALEXISIS++ datasets. Total unique candidate subs. refers to the number of unique candidate substitutions returned from generating k=10 candidate substitutions per context.

then examines both the first unaltered sentence and the words left and right of the [MASK] special token in the altered second sentence. It uses this information to predict a candidate substitution for the masked complex word. From this, an LLM is able to predict a candidate substitution that is suitable for both the provided context and for replacing the complex word. Qiang et al. (2020) was the first to apply MLM for Spanish SG. Their LSBert model surpassed all prior state-of-the-art approaches (Paetzold and Specia, 2017b), including the use of lexicon, rule-based, statistical, n-gram, and word embedding models (Paetzold and Specia, 2017b). LSBert was used as the baseline model at the TSAR-2022 shared-task (Saggion et al., 2022). Inspired by the performance of LSBert, other studies have subsequently used MLM for SG (Ferres and Saggion, 2022; North et al., 2022a; Whistely et al., 2022; Wilkens et al., 2022).

**Available Resources** A number of LS datasets containing complex words in context with gold candidate substitutions are available (North et al., 2022c). For English, there are LexMTurk (Horn et al., 2014) with 500 complex words, BenchLS (Paetzold and Specia, 2016a) with 929 complex words, and NNSeval (Paetzold and Specia, 2016b) with 239 complex words. For other languages, there is EASIER (Alarcón et al., 2021) with 5,310 Spanish complex words, SIMPLEX-PB (Hartmann and Aluísio, 2020) with 730 Portuguese complex words, and HanLS (Qiang et al., 2021) with 534 Chinese complex words. There are also datasets that contain a large number of complex words in context without gold candidate substitutions (Yimam et al., 2018; Maddela and Xu, 2018; Shardlow et al., 2020, 2022). The largest LS dataset that contains both context and gold candidate substitutions is ALEXISIS, referring to the combined English, Spanish (ALEXISIS-ES) (Ferres and Saggion, 2022), and Portuguese (ALEXISIS-PT) (North et al., 2022b) dataset used at the TSAR-2022 shared-task.

### 3 ALEXISIS+

As detailed in Section 2, MLM requires context in order to predict a suitable candidate substitution for a given complex word. Furthermore, MLM also requires a set of gold candidate substitutions to evaluate the quality of those it produces. With this in mind, we expand the ALEXISIS dataset by including a large number of unique additional contexts (Table 1). We then use these additional contexts to produce alternative candidate substitutions through MLM that differ from those generated solely on the original ALEXISIS dataset with examples of these alternative candidate substitutions being provided in Table 2. As such, we introduce ALEXISIS+ and ALEXISIS++, two large expansions of the original ALEXISIS dataset that allow for an IR approach to SG and SS, and that demonstrate how the collection of additional contexts can be used to improve LS performance under the same conditions of the TSAR-2022 shared-task (Saggion et al., 2022).

ALEXISIS+ and ALEXISIS++ were constructed using only the data made available to the participants of the TSAR-2022 shared-task (Saggion et al., 2022). We retrieve instances from the Common-Crawl News (CC-News) dataset<sup>2</sup> by searching for the 386 English, 381 Spanish, and 386 Portuguese complex words given to the original participants of TSAR-2022. The CC-News dataset contains crawled data from news articles all over the world. We restricted our search to news articles with domain urls that contained either one of the following: *.uk*, *.usa* or *.com* for English, *.es*, *.mx*, *.ve*, *.pes*, *.cl*, or *.ec* for Spanish, and *.pt* or *.br* for Portuguese. In this way, we reduced the likelihood of articles containing multiple languages, and we were able to make sure that each context was in the same language as the searched for complex word. Those contexts which contained a match with the original complex word were then extracted. No additional data pre-processing or cleaning was conducted on

<sup>2</sup>CC-News: <https://data.commoncrawl.org/crawl-data/CC-NEWS/index.html>

Lang.	Complex Word	Data	Type	Sentences with same Complex Word	Generated Candidate Subs. (Word.Sim)	Sent.Sim.
EN	<u>replica</u>	A	<b>Original</b>	The statue was moved to the Academia, Gallery and later replaced... by a <u>replica</u> .	<b>duplicate</b> (0.398), replacement (0.333), restoration (0.286), statue (0.426), ...	0.308
		A+	<b>Additional</b>	His project that he chose an exact day and time for the <u>replica</u> he created....	<b>copy</b> (0.503), version (0.322) prototype (0.518), clone (0.456), ...	
ES	<u>municipio</u>	A	<b>Original</b>	Cobisa es un <u>municipio</u> español de la.. [Cobisa is a Spanish municipality in the]..	<b>pueblo</b> (0.5143), ayuntamiento (0.750), localidad (0.691), barrio (0.561), ..	0.490
		A+	<b>Additional</b>	El tortuga reapareció en el <u>municipio</u> ... [The turtle reappeared in the municipality]..	<b>pedanía</b> (0.542), pedanías (0.542), barriada (0.501), huerta (0.276), ...	
PT	<u>incremento</u>	A	<b>Original</b>	Coronel reconheceu <u>incremento</u> roubos... [Colonel acknowledged <u>increased</u> robberies]..	<b>crecimento</b> (0.856), aumento (0.878), incre (0.835), avanço (0.680), ...	0.585
		A+	<b>Additional</b>	Projetos inscritos devem... <u>incremento</u> ... [Submitted projects must... <u>increase</u> ]..	<b>ativos</b> ( 0.505), relevantes (0.541), diversos (0.407), essenciais (0.507), ...	

Table 2: Example instances including original and additional sentences (contexts) and candidate substitutions taken from the ALEXSIS (A) and ALEXSIS+ (A+) datasets. Generated candidate substitutions were produced via MLM per Section 3 with the best candidate substitution being shown in bold. Complex words are underlined and translations shown in [...]. Only snapshots of the sentences are provided. The sentence similarity (Sent.Sim) and word similarity (Word.Sim) between the additional and original sentence embeddings and the embedding of the complex word are also shown.

the extracted contexts.

ALEXSIS+ has a total of 12,831, 13,353, and 13,541 matched complex words in unique contexts for English, Spanish, and Portuguese, respectively. The larger ALEXSIS++ dataset contains matched complex words in 33,149 unique contexts only for English, including those contexts already provided by ALEXSIS+. Both datasets provide embedding similarity scores between their additional sentences and the original context (Sent.Sim), as well as between their additional candidate substitutions and the original complex word (Word.Sim). Sentence embeddings were generated using Sentence-BERT (SBert) (Reimers and Gurevych, 2019). SBert is a state-of-the-art sentence-encoder. It employs siamese and triplet network structures to produce sentence embeddings that can be used to compare the semantic similarity between sentences by calculating the cosine similarity between sentence embeddings. English word embeddings were obtained using the *en-vectors-web-lg* model that provides ~500k word vectors. Spanish and Portuguese word embeddings were taken from the *pt-core-news-lg*, and *es-core-news-lg* models trained on crawled news articles.

**Dataset Format** ALEXSIS+ and ALEXSIS++ are divided into three sub-corpora corresponding to the three languages, English (EN), Spanish (ES), and Portuguese (PT). Each dataset contains: original CW, context, and candidate substitutions from the TSAR-2022 shared-task, new contexts and new candidate substitutes generated on each new context, cosine similarities between new and old contexts and word similarities between word embeddings of the new candidate substitutions and the

target complex word. ALEXSIS+ and ALEXSIS++ have the following nine headers separated by tab ( $\backslash$ t):

1. **ID**: instance id that is made up of the original instance id (e.g. 01) and the new additional context id. (e.g. 104): 01-104.
2. **ALEXSIS.CW**: the original complex word taken from ALEXSIS and used at TSAR-2022.
3. **ALEXSIS.Context**: the original context for the given complex word taken from ALEXSIS and used at TSAR-2022.
4. **Candidate.Subs@n**: the candidate substitutions generated using MLM on the instances provided by TSAR-2022.
5. **Additional.Context**: new additional context obtained from the CC-News dataset.
6. **Additional.Subs@n**: new additional candidate substitutions generated using MLM on the additional contexts taken from the CC-News dataset.
7. **Sent.Sim**: the cosine similarities between the SBert sentence embedding of the additional context and the original context provided by TSAR-2022.
8. **Word.Sim**: : the cosine similarities between the word embeddings of the additional candidate substitutions and the original complex word provided by TSAR-2022.
9. **Gold.Labels**: the original gold candidate substitutions provided by TSAR-2022.

## 4 Approach

### 4.1 Substitute Generation

We experimented with three pre-trained LLMs trained with a MLM objective. Following the results of [Ferres and Saggion \(2022\)](#) and [North et al. \(2022a\)](#), we chose three monolingual rather than multilingual LLMs given their superior performance for language-specific SG ([Saggion et al., 2022](#)). We use ELECTRA ([Clark et al., 2020](#)) for English, RoBERTa-large-BNE ([Fandiño et al., 2022](#)) for Spanish, and BERTimbau ([Souza et al., 2020](#)) for Portuguese. ELECTRA was pre-trained on English Wikipedia data with a vocabulary size of 30,522 tokens. RoBERTa-large-BNE was pre-trained on the National Library of Spain corpus ([Fandiño et al., 2022](#)) that consists of 135 billion Spanish tokens scraped from Spanish websites. BERTimbau was pre-trained on the Brazilian Web as Corpus ([Wagner Filho et al., 2018](#)) that contains 2.7 billion Portuguese tokens scraped from Brazilian websites.

Figure 2 outlines our approach. We used our MLM models to generate  $k = 10$  candidate substitutions for each masked complex word in context taken from the original TSAR-2022 dataset (ALEXsis) as well as ALEXsis+ or ALEXsis++. Those candidate substitutes generated by the additional contexts provided by ALEXsis+ or ALEXsis++ were subject to several SS filters or steps. If a candidate substitution managed to pass these SS filters, then that candidate substitution would be used instead of the previous candidate substitution generated on the original ALEXsis dataset (Figure 2). We explain each SS filter in the following section.

### 4.2 Substitute Selection

A total of five different SS filters were applied to the candidate substitutions generated by the additional contexts of ALEXsis+ or ALEXsis++. These filters were inspired by well-established methods of SS, including the use of WordNet ([Fellbaum, 2010](#)), semantic similarity between word embeddings (EmbeddingSim) and word length ([Paetzold and Specia, 2017a](#)), as well as recent advances in deep learning, such as chain-of-thought prompting ([Aumiller and Gertz, 2022](#); [Vásquez-Rodríguez et al., 2022](#)). These different SS filters have been used in different experimental pipeline setups, as later described in Section 4.3.

**WordNet+EmbeddingSim** WordNet was used to calculate the similarity between a candidate substitution and the original complex word. The returned similarity score was used alongside the cosine similarity produced by comparing the word embedding of a candidate substitution and the original complex word. These word embeddings were generated by the language models described in Section 3 and were dependent on the language. Early experiments on the ALEXsis+ dataset were conducted to identify optimum threshold values for both word similarity metrics. Similarity values between 0.55 and 0.65 were found to produce the highest number of candidate substitutions from the additional contexts that went on to replace the original candidate substitution, regardless of language. Interestingly, WordNet’s limited vocabulary was seen to aid this filtering process since out-of-vocabulary words that may have been problematic were automatically removed from the list of potential candidate substitutions.

**WordFreq** Zipf’s Law suggests that words with lower frequency in a text tend to be longer and thus can be seen as more complex than words that appear more often and are shorter ([Quijada and Medero, 2016](#); [Desai et al., 2021](#)). We subsequently used word frequency as a second initial SS filter during our early experiments. Those candidate substitutions which had been generated from the additional contexts more than twice passed this filter, whereas those with a generated frequency of less than two were removed.

**EmbeddingSim** Later SS approaches required that a greater number of candidate substitutions passed the initial filters. As such, the WordNet Lin similarity and WordFreq thresholds from our initial experiments were dropped. However, we maintained a cosine similarity of 0.5 between the word embedding of a candidate substitution and the original complex word. We named this SS filter: EmbeddingSim (EmbSim).

**PromptLearning** Prompt learning (PromptL) is a new state-of-the-art technique used for LS ([Aumiller and Gertz, 2022](#); [Vásquez-Rodríguez et al., 2022](#)). It involves feeding input into a LLM, referred to as a prompt or set of prompts, that both describe the task and are worded in such a way as to elicit a desired output. For instance, we fed three prompts into a GPT-3 ([Brown et al., 2020](#)) model that were designed to identify three viable candi-



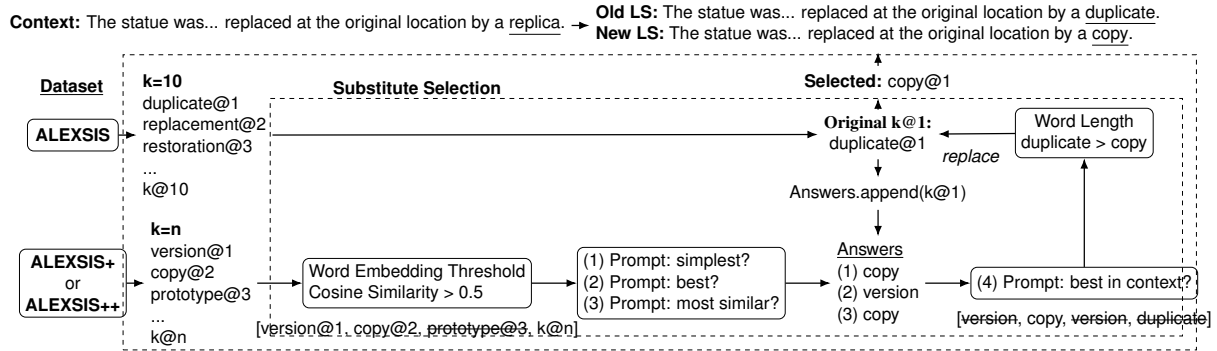


Figure 2: Our second IR approach (**pipeline b**) for LS via MLM using the ALEXISIS+ and ALEXISIS++ datasets. This approach is responsible for the results shown in Table 3.  $n$  being the number of additional candidate substitutions produced for a given a complex word from the ALEXISIS dataset using a different sentence from the ALEXISIS+ or ALEXISIS++ datasets.

date substitutions from a list of potential candidates returned from previous filters.

1. **Prompt:** What word is the *simplest* replacement for <Complex.Word> in this list?
2. **Prompt:** What word is the *best* replacement for <Complex.Word> in this list?
3. **Prompt:** What word is the *most similar* word to <Complex.Word> in this list?

The GPT-3 model then selects a maximum of one candidate substitution which best answers each of these prompts. The outputted three candidate substitutions are then appended to a new list, whereby the previous candidate substitution generated from the original ALEXISIS dataset is also appended. The model is then fed one final prompt:

4. **Prompt:** Given the *above context*, what is the *best replacement* for <Complex.Word> in this list?

This fourth prompt is able to determine out of the *simplest*, *best*, and *most similar* candidate substitution to the complex word, which is the best fit in the complex word’s provided context. Through such chain-of-thought prompting, we are able to deduce the most appropriate candidate substitution for a given context and complex word from those generated from all of the additional contexts in the ALEXISIS+ and ALEXISIS++ datasets.

**WordLength** We used word length as an additional SS filter. This SS filter was also inspired by Zipf’s Law. It was applied to the candidate substitution returned from our prompt learning SS filter. If the returned candidate substitution generated was

greater in length than the original candidate substitution generated from the ALEXISIS dataset, then it is removed and the original candidate substitution is put forward. If, however, said additional candidate substitution is shorter, then it was used to replace the original candidate substitution and sent to our final filter.

**BertEmbSim** Our final filter used the pre-trained word embeddings from the BERT model to compute the cosine similarity of the complex word with the original candidate substitution ( $\text{cos\_old}$ ), and the cosine similarity of the complex word with the new candidate substitution ( $\text{cos\_new}$ ) generated from ALEXISIS+ or ALEXISIS++. If  $\text{cos\_old}$  was greater than  $\text{cos\_new}$ , and if the absolute value of the difference between the two was more than 10%, we used the original candidate substitution, else we returned the new candidate substitution.

### 4.3 Substitute Selection Pipeline

We experimented with three combinations of the above SS filters which resulted in three SS pipelines. These SS pipelines, (a). to (c)., are described below.

**Pipeline (a).** This SS pipeline was used during early experiments. Candidate substitutions produced by the additional contexts were subject to two WordNet Lin and cosine word embedding similarity thresholds both set to 0.5. Candidate substitutions that passed these threshold were then subject to a word frequency check ( $>2$ ) and a word length check ( $<$ original candidate substitution) before replacing the original candidate substitution.

**Pipeline (b).** This SS pipeline is depicted in Figure 2. It is responsible for the results shown in

Lang.	Source	Size	ACC	MAP	POT
EN	ALEXSIS++	33,149	<b>0.495</b>	<b>0.495</b>	<b>0.495</b>
	ALEXSIS+	12,831	0.479	0.479	0.479
	ALEXSIS	374	0.484	0.484	0.484
ES	ALEXSIS+	13,353	<b>0.110</b>	<b>0.138</b>	<b>0.138</b>
	ALEXSIS	368	0.108	0.135	0.135
PT	ALEXSIS+	13,541	<b>0.479</b>	<b>0.489</b>	<b>0.489</b>
	ALEXSIS	374	0.476	0.487	0.487

Table 3: Performance of ALEXSIS, ALEXSIS+, and ALEXSIS++ when utilized by the same model for SG and evaluated on k@1 candidate substitution. Performances were evaluated on the original TSAR-2022 test set. Best performances are shown in **bold**.

Table 3. We dropped the lin similarity threshold produced by WordNet to increase the number of candidate substitutions passed to later SS filters. However, the same cosine word embedding similarity threshold of 0.5 was maintained. Additional candidate substitutions were then filtered by applying prompt learning. The first round of prompt learning reduces the list of potential candidate substitutions to three. The original candidate substitution generated from the ALEXSIS dataset is then added to this list. The last round of prompt learning selects only one out of the now four candidate substitutions. The returned candidate substitution is then subjected to a final word length check (<original candidate substitution).

**Pipeline (c).** After conducting the majority of our experiments, we discovered several occasions whereby the additional candidate substitution selected by our prompt learning SS filter was unsuitable for the given context (Section 5.1). To account for this, we applied an additional cosine similarity threshold between BERT produced word embeddings (BertEmbSim). All other SS filters are the same as SS pipeline (b) shown in Figure 2.

## 5 Evaluation

This section evaluates the performance of Electra, RoBERTa-large-BNE, and BERTimbau on the TSAR-2022 test set using our IR approach to SG and SS and the ALEXSIS+ and ALEXSIS++ datasets (Section 5.1). We also provide the performance of our various SS pipelines (Section 5.3). For the evaluation, we removed duplicate gold labels within the TSAR-2022 test set. Performances are reported in terms of accuracy (ACC), mean absolute precision, and potential following the TSAR-2022 shared-task (Saggion et al., 2022).

The performances reported at the TSAR-2022 shared-task (Section 2) show that LS is still chal-

lenging. Even small improvements in performances can lead to greater gains down-stream for TS. For this reason, LS is often primarily evaluated on the quality of the top candidate substitution produced (k@1). The accuracy of the top k@1 candidate substitution (ACC@1) is the ratio of instances whereby the best candidate generated is also the most appropriate candidate substitution among the gold labels. ACC@1 is often used to determine the overall performance of a LS system, since it is this candidate substitution which replaces the complex word. In addition, LS is also evaluated on its F1-score, potential (POT) and mean average precision (MAP). POT is the ratio of the candidate substitutions that are within all of the gold labels. MAP provides a score of the number of the returned candidate substitutions which match a gold label and its index.

### 5.1 ALEXSIS+ Performance

Our Spanish (RoBERTa-large-BNE) and Portuguese (BERTimbau) models benefited from the additional candidate substitutions provided by ALEXSIS+ (Table 3). RoBERTa-large-BNE’s k@1 candidate substitutions increased in accuracy going from an ACC@1 score of 0.108 to 0.110, BERTimbau’s final candidate substitutions saw an almost identical increase in its ACC@1, increasing from 0.476 and 0.479. However, this increase did not apply to our English (ELECTRA) model.

There are two possible causalities for this irregular improvement. The first was recognized when examining BERTimbau’s MAP@3 score: 0.292, after having generated three (k@3) rather than one candidate substitution. This score is superior to that achieved by using only the original ALEXSIS dataset which obtained a MAP@3 of 0.290 when likewise generating the same number of candidate substitutions. MAP evaluates the quality of the candidate substitution produced in compari-

Lang.	Source	Approach		Top-k=1 (@1)			Top-k=3 (@3)		
		SG	SS: Step1→Step2→Step3→Step4	ACC	MAP	POT	ACC	MAP	POT
EN	ALEXISIS++	MLM	(c). EmbSim→PromptL→WordLen→BertEmbSim	<b>0.495</b>	<b>0.495</b>	<b>0.495</b>	<b>0.765</b>	<b>0.337</b>	<b>0.765</b>
			(b). EmbSim→PromptL→WordLen	<b>0.495</b>	<b>0.495</b>	<b>0.495</b>	0.757	0.329	0.757
			(a). WordNet+EmbedSim→WordFreq→WordLen	0.487	0.487	0.487	0.733	0.335	0.733
EN	ALEXISIS	MLM	None	0.484	0.484	0.484	0.738	0.336	0.738

Table 4: Shows performances of various SS approaches applied to the additional candidate substitutions generated by ALEXISIS++ and evaluated on the original TSAR-2022 test set. Best performances are shown in **bold**.

son to the gold labels as well as its positional rank (Section 5). From this, we can infer that the use of ALEXISIS+ has resulted in an original candidate substitution at rank 2 or 3 being moved to a rank 1 position. This would explain BERTimbau’s improved ACC@1, since it’s k@1 candidate substitution is now more aligned with the k@1 candidate substitutions within the TSAR 2022 test set’s gold labels. The second feasible causality may be the fourth prompt within our prompt learning SS filter. Previously mentioned in Section 4.3, we discovered several occasions whereby the returned additional candidate substitution was unsuitable for the given context. Take the following complex word in context (a), and the simplifications produced by using the original (old) and additional (new) candidate substitutions as an example.

- (a) **Complex:** “There’s **conflicting** evidence about whether sick ants actually smell different from healthy ones or not.”
- (b) **Old LS:** “There’s **mixed** evidence about whether sick ants actually smell different from healthy ones or not.”
- (c) **New LS:** “There’s **some** evidence about whether sick ants actually smell different from healthy ones or not.”

The additional candidate substitution: “*some*” returned from our prompt learning SS filter, and used in the generated (new) simplification, may be considered to be simpler in comparison to the original candidate substitution: “*mixed*”. Nevertheless, in this context “*mixed*” is the more suitable candidate. This is because it is more semantically similar to the complex word “*conflicting*”. GPT-3 has, therefore, failed to select the most appropriate candidate substitution after having received our fourth context orientated prompt (Section 4.3). ALEXISIS++ and the additional BERT Embedding Similarity threshold (BertEmbSim) were created to overcome this issue by either supplying more candidate substitutions or by improving the performance of our

SS pipeline (b). The following sections provide model performances on ALEXISIS++ (Section 5.2) as well as performances before and after incorporating the BertEmbSim SS filter (Section 5.3).

## 5.2 ALEXISIS++ Performance

The additional contexts provided by ALEXISIS++ improved the quality of the candidate substitutions selected by our approach (Figure 2). These additional contexts allowed for the generation of more high quality candidate substitutions through MLM. A total of 289,379 additional candidate substitutions were provided surpassing the 120,645 produced by ALEXISIS+. As a result, increases in performances were recorded across all metrics for our English (ELECTRA) model. ACC@1, POT@1, and MAP@1 rose to 0.495 from 0.479, respectively. Despite increasing in performances being small, it is clear that the use of ALEXISIS++ is able to further increase LS performance beyond that achieved by the ALEXISIS and ALEXISIS+ datasets. It is, therefore, highly likely that the degree of improvement caused by our IR approach positively correlates with the number of additional contexts it takes into consideration going from 386 for ALEXISIS, 12,831 for ALEXISIS+, to 33,149 for ALEXISIS++. However, this positive correlation is only realized if an accurate SS pipeline is applied.

## 5.3 BERT Embeddings

We compared the performance of attaching the BertEmbSim SS filter to pipeline (b) against that achieved by our previous SS pipelines and LS performance without SS (Table 4). It was found that this new pipeline (c) outperformed all of our previous methods of SS for English when set to produce three candidate substitutions (k@3). The use of the BertEmbSim SS filter (c) saw an increase in ACC@3 of 0.765 from 0.757 in comparison to our previous pipeline (b). This coincided with improvements in MAP and POT scores, with a MAP@3 and POT@3 also rising to 0.337 from 0.329 and

0.765 from 0.757, respectively. In addition, having no SS filter achieved an inferior ACC@3 and POT@3 of 0.738 and 0.738, respectively, when compared to pipelines (b) and (c).

The BertEmbSim SS filter (c) was seen to produce candidate substitutions that were more suited for a complex word’s context than in comparison to the previous prompt learning filter (b). This was the case for the previous example shown in Section 5.2, as the BertEmbSim SS filter was able to correctly identify “*mixed*” as being a more appropriate candidate substitution for the complex word “*conflicting*” than compared to the additional candidate substitution “*some*”. In this instance, the cosine similarity between BERT word embeddings of a candidate substitution and a complex word has, thus, exceeded GPT-3’s ability at determining the most appropriate replacement for a given context. This explains the superior performance of our BertEmbSim SS filter (c).

## 6 Conclusion and Future Work

This paper presents ALEXSIS+ and ALEXSIS++, two new version of the ALEXSIS dataset used at the TSAR-2022 shared-task (Saggion et al., 2022). These datasets contain more than 50,000 unique sentences covering three languages retrieved from news corpora and annotated with cosine similarities to the original complex word and sentence.

We have demonstrated that the use of these datasets, alongside an effective method of SS, can be used to generate and then select a more appropriate candidate substitution which, in turn, improves LS performance without the need for re-training or fine-tuning. In other words, results showed that the use of additional unique contexts can result in increases in LS performance, despite these contexts being dissimilar from the original context of the complex word. This increase in performance may appear small. However, even a small improvement in LS can have wider downstream implications that enhance the performance of a TS system substantially. We hypothesize that through further experimentation with alternative SG methods and SS filters, the performance gained by using ALEXSIS+ and ALEXSIS++ will increase. We provide these two new LS datasets and make them publicly available to the wider research community.

To the best of our knowledge, this is the first IR approach to LS opening exciting new avenues for research in this field. We show that the approach

increases overall performance and that it can be applied to any LS model or language. In the future, we would like to incorporate this IR-based approach in a real-world personalized TS system that can be used in educational technology applications and online learning (McCarthy et al., 2022).

## Acknowledgment

We would like to thank the creators of the ALEXSIS datasets for making the datasets available for our researcher. We further thank the anonymous BEA reviewers for their insightful feedback.

## References

- Rodrigo Alarcón, Lourdes Moreno, and Paloma Martínez. 2021. Lexical Simplification System to Improve Web Accessibility. *IEEE Access*.
- Dennis Aumiller and Michael Gertz. 2022. UniHD at TSAR-2022 Shared Task: Is Compute All We Need for Lexical Simplification? In *Proceedings of TSAR*.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, and Others. 2020. Language Models Are Few-Shot Learners. In *Proceedings of NeurIPS*.
- Kevin Clark, Minh-Thang Luong, Quoc Le, and Christopher Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *Proceedings of ICLR*.
- Abhinandan Desai, Kai North, Marcos Zampieri, and Christopher Homan. 2021. LCP-RIT at SemEval-2021 Task 1: Exploring Linguistic Features for Lexical Complexity Prediction. In *Proceedings of SemEval*.
- Asier Gutiérrez Fandiño, Jordi Armengol Estapé, Marc Pàmies, Joan Llop Palao, Joaquin Silveira Ocampo, and Others. 2022. Maria: Spanish language models. *Procesamiento del Lenguaje Natural*, 68.
- Christiane Fellbaum. 2010. Wordnet. In *Theory and applications of ontology: computer applications*, pages 231–243. Springer.
- Daniel Ferres and Horacio Saggion. 2022. ALEXSIS: A dataset for lexical simplification in Spanish. In *Proceedings of LREC*.
- Nathan Siegle Hartmann and Sandra Maria Aluísio. 2020. Adaptação lexical automática em textos informativos do português brasileiro para o ensino fundamental. *Linguamática*, 12(2):3–27.
- Colby Horn, Cathryn Manduca, and David Kauchak. 2014. Learning a lexical simplifier using Wikipedia. In *Proceedings of ACL*.

- Mounica Maddela and Wei Xu. 2018. A word-complexity lexicon and a neural readability ranking model for lexical simplification. In *Proceedings of EMNLP*.
- Kathryn S McCarthy, Scott A Crossley, Kayla Meyers, Ulrich Boser, Laura K Allen, Vinay K Chaudhri, Kevyn Collins-Thompson, Sidney D’Mello, Munmun De Choudhury, Kumar Garg, et al. 2022. Toward more effective and equitable learning: Identifying barriers and solutions for the future of online education. *Technology, Mind, and Behavior*, 3(1).
- Kai North, Alphaeus Dmonte, Tharindu Ranasinghe, and Marcos Zampieri. 2022a. GMU-WLV at TSAR-2022 Shared Task: Evaluating Lexical Simplification Models. In *Proceedings of TSAR*.
- Kai North, Tharindu Ranasinghe, Matthew Shardlow, and Marcos Zampieri. 2023. Deep learning approaches to lexical simplification: A survey. *arXiv preprint arXiv:2305.12000*.
- Kai North, Marcos Zampieri, and Tharindu Ranasinghe. 2022b. ALEXSIS-PT: A new resource for portuguese lexical simplification. In *Proceedings of COLING*.
- Kai North, Marcos Zampieri, and Matthew Shardlow. 2022c. Lexical Complexity Prediction: An Overview. *ACM Computing Surveys*.
- Gustavo Paetzold and Lucia Specia. 2017a. Lexical simplification with neural ranking. In *Proceedings of ACL*.
- Gustavo H. Paetzold and Lucia Specia. 2017b. A Survey on Lexical Simplification. *Journal of Artificial Intelligence Research*, 60(1):549–593.
- Gustavo Henrique Paetzold and Lucia Specia. 2016a. Benchmarking lexical simplification systems. In *Proceedings of LREC*.
- Gustavo Henrique Paetzold and Lucia Specia. 2016b. Unsupervised lexical simplification for non-native speakers. In *Proceedings of AACL*.
- Jipeng Qiang, Yun Li, Zhu Yi, Yunhao Yuan, and Xindong Wu. 2020. Lexical simplification with pre-trained encoders. In *Proceedings of AACL*.
- Jipeng Qiang, Xinyu Lu, Yun Li, Yunhao Yuan, and Xindong Wu. 2021. Chinese Lexical Simplification. *IEEE Press*, 29:1819–1828.
- Maury Quijada and Julie Medero. 2016. HMC at SemEval-2016 Task 11: Identifying Complex Words Using Depth-limited Decision Trees. In *Proceedings of SemEval*.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of EMNLP-IJCNLP*.
- Horacio Saggion, Sanja Štajner, Daniel Ferrés, Kim Cheng Sheang, Matthew Shardlow, Kai North, and Marcos Zampieri. 2022. Findings of the TSAR-2022 Shared Task on Multilingual Lexical Simplification. In *Proceedings of TSAR*.
- Matthew Shardlow, Michael Cooper, and Marcos Zampieri. 2020. CompLex — a new corpus for lexical complexity prediction from Likert Scale data. In *Proceedings of READI*.
- Matthew Shardlow, Richard Evans, and Marcos Zampieri. 2022. Predicting lexical complexity in english texts: the complex 2.0 dataset. *Language Resources and Evaluation*, 56:1153–1194.
- Jiayin Song, Jingyue Hu, Leung-Pun Wong, Lap-Kei Lee, and Tianyong Hao. 2020. A New Context-Aware Method Based on Hybrid Ranking for Community-Oriented Lexical Simplification. In *Proceedings of DASFAA*.
- Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. 2020. BERTimbau: pretrained BERT models for Brazilian Portuguese. In *Proceedings of BRACIS*.
- Sanja Štajner, Daniel Ferrés, Matthew Shardlow, Kai North, Marcos Zampieri, and Horacio Saggion. 2022. Lexical Simplification Benchmarks for English, Portuguese, and Spanish. *Frontiers in Artificial Intelligence*.
- Laura Vásquez-Rodríguez, Nhung Nguyen, Sophia Ananiadou, and Matthew Shardlow. 2022. UoM&MMU at TSAR-2022 Shared Task: Prompt Learning for Lexical Simplification. In *Proceedings of TSAR*.
- Jorge A. Wagner Filho, Rodrigo Wilkens, Marco Idiart, and Aline Villavicencio. 2018. The brWaC corpus: A new open resource for Brazilian Portuguese. In *Proceedings of LREC*.
- Peniel John Whistely, Sandeep Mathias, and Galiveeti Poornima. 2022. PresiUniv at TSAR-2022 Shared Task: Generation and Ranking of Simplification Substitutes of Complex Words in Multiple Languages. In *Proceedings of TSAR*.
- Rodrigo Wilkens, David Alfter, Rémi Cardon, Isabelle Gribomont, and Others. 2022. CENTAL at TSAR-2022 Shared Task: How Does Context Impact BERT-Generated Substitutions for Lexical Simplification? In *Proceedings of TSAR*.
- Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri. 2018. A Report on the Complex Word Identification Shared Task 2018. In *Proceedings of BEA*.

# Generating Better Items for Cognitive Assessments Using Large Language Models

Antonio Laverghetta Jr. and John Licato

University of South Florida

Department of Computer Science and Engineering

Tampa, FL, USA

{alaverghett, licato}@usf.edu

## Abstract

Writing high-quality test questions (items) is critical to building educational measures but has traditionally also been a time-consuming process. One promising avenue for alleviating this is automated item generation, whereby methods from artificial intelligence (AI) are used to generate new items with minimal human intervention. Researchers have explored using large language models (LLMs) to generate new items with equivalent psychometric properties to human-written ones. But can LLMs generate items with *improved* psychometric properties, even when existing items have poor validity evidence? We investigate this using items from a natural language inference (NLI) dataset. We develop a novel prompting strategy based on selecting items with both the best and worst properties to use in the prompt and use GPT-3 to generate new NLI items. We find that the GPT-3 items show improved psychometric properties in many cases, whilst also possessing good content, convergent and discriminant validity evidence. Collectively, our results demonstrate the potential of employing LLMs to ease the item development process and suggest that the careful use of prompting may allow for iterative improvement of item quality.

## 1 Introduction

AI is having increasingly profound impacts on educational and psychological measurement (Chen et al., 2020; Tavast et al., 2022). Technologies built on AI and machine learning, including educational data mining (Romero and Ventura, 2020), intelligent tutoring systems (Mousavinasab et al., 2021), deep item response theory (Cheng et al., 2019), and deep knowledge tracing (Piech et al., 2015), among others (Asfahani, 2022, inter-alia) are transforming educational and psychological measurement, and this trend seems likely to continue.

One promising educational application of large language models (LLMs) is for the automatic gen-

eration of test items (AIG). Writing high-quality test items is critical to building effective educational assessments, but has also traditionally been a time-consuming process, as items must be developed by experts and undergo numerous rounds of review (Bandalos, 2018). There has been significant research interest in using AIG to create high-quality items with minimal intervention to speed up the test development process (Prasetyo et al., 2020). Prior work has demonstrated that LLMs can generate items with at least face validity (i.e. they *appear* valid based on item content) for both non-cognitive (Götz et al., 2023) and cognitive (Attali et al., 2022) constructs. Careful psychometric analysis of items generated from such models has also revealed that they are just as valid and reliable as their human written counterparts (Lee et al., 2023). Although promising, this research has largely focused on generating items for constructs that have been well-studied, using items already known to have strong validity evidence. Suppose an educator wishes to develop a test for a new construct where existing items may have only undergone pretesting. Or suppose the educator wishes to use a new type of item for a well-established domain (e.g. a test of algebraic reasoning that uses a novel item format). In either case, the items will likely have limited validity evidence, and much time would need to be spent revising the items to improve their psychometric properties before they can be used.

In this work, we ask: can LLMs be used to generate valid and reliable items even in these scenarios where existing items have only limited validity evidence? If so, LLM-based AIG could be used to iteratively improve the psychometric properties of items, explore the underlying construct space, and shed light on what makes a good item.

We explore this using GPT-3 (Brown et al., 2020) and focus on generating items that test for natural language inference (NLI) (Dagan et al., 2006; Bowman et al., 2015). NLI is an important cognitive

construct in NLP research which, to our knowledge, has only undergone limited psychometric analysis in human participants (Laverghetta Jr. et al., 2021). We develop a novel prompting strategy that uses the psychometric properties of items, calculated using prior human responses, to select the most informative examples to send to the model to maximize the quality of the generated examples. **Our main contributions are as follows:**

1. We develop a novel prompting strategy for generating items by selecting items to include as context based on the psychometric properties they possess, focusing primarily on item discrimination.
2. Using GPT-3 we test our approach using the GLUE broad coverage diagnostic (Wang et al., 2018), a popular cognitive task in NLP research. We perform an extensive analysis of the psychometric properties of the generated items and find that those from GPT-3 show stronger evidence for validity and reliability than those written by humans in most cases.

## 2 Related Work

### 2.1 Automated Item Generation

Psychometricians have explored how to automate item generation for decades (Prasetyo et al., 2020). Early attempts focused on developing item models, which are systems that can interchange certain keywords in the item while keeping other parts of it constant (Bejar et al., 2002). While item models are theoretically justified and very likely to produce psychometrically valid items, developing them requires a great deal of manual effort, as both the item stem and other components must still be manually written. Furthermore, item models are limited in the diversity of content they can generate. These drawbacks have motivated recent work to investigate using LLMs as the item generator. von Davier (2018) was one of the first to explore this and used recurrent neural networks to generate items for a personality assessment. The advent of the transformer (Vaswani et al., 2017; Devlin et al., 2019; Brown et al., 2020) led to the creation of LLMs which could generate much more coherent and semantically accurate text, leading to further interest in LLM-based AIG. Götz et al. (2023) generated a large number of personality items using GPT-2 (Radford et al., 2019), and showed that at least some of these items passed face validity checks.

Maertens et al. (2021) developed a test for misinformation susceptibility, using LLM-generated items. Hernandez and Nie (2022) developed a system for the automatic generation and validation of test items, using autoregressive LLMs for generation and autoencoding LLMs for validation. Lee et al. (2023) extensively evaluated the psychometric properties of GPT-3 generated personality items, including analysis of internal structure, differential item functioning, and reliability. They concluded that the validity evidence for machine-generated items was just as strong, if not stronger than, for human-written ones. While much work has focused on non-cognitive assessments, others have explored LLM-based AIG for educational assessments. Notably, Chan et al. (2022) used the BERT (Devlin et al., 2019) LLM to generate grammar reading exercises. Zou et al. (2022) and Rathod et al. (2022) used transformers to generate true/false and reading comprehension questions. Attali et al. (2022) used transformer-based LLMs to generate items for the Duolingo English Test. Zu et al. (2023) used a combination of finetuning and prompt-based learning to train GPT-2 to generate distractors for fill-in-the-blank vocabulary items. A common theme throughout these works is the focus on well-studied assessments, and the use of items that have already been psychometrically validated in the prompt. Their goal is thus to generate items that maintain existing psychometric properties, which is different from our goal of generating items with *improved* properties.

### 2.2 Synthetic Data Generation in NLP

When it comes to gathering high-quality data, NLP researchers have concerns that overlap with those faced by the measurement community. Training examples for popular NLP tasks, including NLI (Bowman et al., 2015), and question answering (QA) (Rajpurkar et al., 2016), have historically been created using crowd-sourced annotations, which is both expensive and time-consuming. The incredibly rapid progress of LLMs in recent years also means that many once challenging datasets quickly become outdated as new models are developed (Ott et al., 2022). There has been significant research interest in using LLMs to generate synthetic training data, forgoing the need to run annotation studies (Schick and Schütze, 2021). Prior work has explored LLM-based data augmentation for QA (Duan et al., 2017), paraphrase identification

(Nigohjkar and Licato, 2021), and NLI (Liu et al., 2022). Typically, this line of research relies on information-theoretic metrics of item quality, for example, dataset maps (Swayamdipta et al., 2020) to evaluate the newly generated items. Most relevant to our work is the study by Liu et al. (2022), who developed a system for using GPT-3 to automatically generate NLI items. However, their approach does not employ methods of assessing validity and reliability commonly used in educational measurement and instead relies on information-theoretic measures of item quality. Our goal is to generate items with improved validity and reliability in both human and LLM populations, using the psychometric properties of the items as the optimization target.

### 3 Generation of Test Items

The General Language Understanding Evaluation (GLUE) (Wang et al., 2018) is a benchmark designed to measure broad linguistic constructs in LLMs. Included in GLUE is a diagnostic set,  $AX$ ,<sup>1</sup> which is meant to be a challenge set for diagnosing faults in LLMs. Items on  $AX$  are framed as NLI: given a premise ( $p$ ) and hypothesis ( $h$ ), a model must determine whether  $p$  entails, contradicts, or is neutral with respect to  $h$  (Dagan et al., 2006; Bowman et al., 2015). Items were written by NLP experts, inspired by categories taken from the Fra-Cas suite (Cooper et al., 1996), and are based on sentences from a variety of artificial and naturalistic contexts. Wang et al. (2018) reported strong inter-rater reliability when labeling a random sample of  $AX$  items, and  $AX$  has been used successfully to evaluate many new LLMs (Brown et al., 2020; Raffel et al., 2020; Chowdhery et al., 2022), which suggests the diagnostic has good predictive validity. Furthermore, Laverghetta Jr. et al. (2021) previously ran human studies on a subset of items from  $AX$ , targeting those testing for propositional structure (PS), quantifiers (Q), morphological negation (MN), and lexical entailment (LE). Table 1 shows example  $AX$  items from these categories. They found that LLMs strongly predicted item difficulties and inter-item correlations in human responses across these categories, indicating good convergent validity for  $AX$  as a test of reasoning in both populations. Collectively, these results demonstrate a surface level of validity for the  $AX$  items (i.e.,

<sup>1</sup> $AX$  being the notation for the diagnostic on the GLUE leaderboard.

Category	$p$	$h$
PS	The cat sat on the mat.	The cat did not sit on the mat.
LE	The water is too hot.	The water is too cold.
MN	The new console is cheap.	The new console isn't cheap.
Q	Several are available.	All are available.

Table 1: Examples of NLI items from each  $AX$  category. MN and Q items have been trimmed and paraphrased to fit in one line, but still fall into their respective categories.

*face validity*); the items appear to function well in preliminary human studies and have been used successfully to find faults within LLM reasoning, but extensive analysis of their psychometric properties has yet to be performed. This makes  $AX$  a good assessment to use for our experiments, as we want items that have *not* undergone extensive psychometric development, and hence may not have strong validity as measures of the construct in question.

Our goal is to use LLMs to generate new items for  $AX$ , such that the psychometric properties of both the items and the test as a whole are improved. Formally, given an LLM  $M$  and a prompt  $p$  that contains one or more items that have a psychometric property  $\theta$ , we seek to sample new items  $i$  from  $M$  that lead to an improvement in  $\theta$ :<sup>2</sup>

$$i \sim M(p) \mid \theta_i > \theta_p \quad (1)$$

Where  $i$  and  $p$  are assumed to test for the same construct (e.g., NLI). Prior work has demonstrated that when LLMs are given existing items as prompts, they can generate new items that match the construct measured by those items (Liu et al., 2022; Lee et al., 2023). We build on this approach by designing prompts to instruct LLMs to generate new items for a particular construct, that *possess a desired psychometric property*. Figure 1 shows one of the prompts we developed. The model is instructed to generate only items that match the target property, and we use items from only one category at a time. We use item discrimination as the target property in our experiments. Discrimination refers to the ability of an item to separate high from low-ability test takers (Bandalos, 2018) and is computed using the item-to-total correlation (the correlation between the responses to a single item and total scores across all items). An item that is

<sup>2</sup>Note that  $\theta_i > \theta_p$  should be taken to mean that the psychometric properties of  $i$  are improved relative to  $p$ , and not necessarily that they are numerically greater.



```

I need to generate new NLI
items for a given trait.
Here are some examples:
###
Trait: High Discrimination
Items (3):
[ITEMS]
###
Trait: Low Discrimination
Items (3):
[ITEMS]
###
Trait: High Discrimination
New Items (5):

```

Figure 1: Prompt structure using the “simple” prompt format. Additional newlines have been added to keep text within margins.

highly discriminating will predict total scores and thus should be maximized. Our use of discrimination was based on preliminary analysis of the data from Laverghetta Jr. et al. (2021), which indicated that at least one item in every category had negative discrimination. In general, items with negative discrimination are regarded as problematic and possibly erroneous, and should not be included in cognitive assessments (Bandalos, 2018), which makes improving the discrimination of the *AX* items a natural optimization target. We use existing human written items as examples of the desired property in the prompt, selecting the top  $k$  items with the highest discrimination as “high discrimination” and the bottom  $k$  items with the lowest discrimination as “low discrimination”.<sup>3</sup> We set  $k = 3$  in our experiments, as we found larger values caused the difference in discrimination to become negligible. By providing examples of both good and bad items, we hope to teach the model general characteristics of high-quality items.<sup>4</sup>

We use GPT-3 (Brown et al., 2020) as our item generator, given its strong performance across many NLP tasks, the presence of an easy-to-use and inexpensive API, and the success prior work has had in using GPT-3 to generate non-cognitive (Lee et al., 2023) and NLI (Liu et al., 2022) items.

<sup>3</sup>Properties are calculated using SPSS version 28. We use only the categories from Table 1.

<sup>4</sup>Note that our approach has strong conceptual similarities to prior work in few-shot item selection for in-context learning (e.g. Walsh et al., 2022), in that the psychometric properties of the items are essentially used to select which shots to use.

We set temperature to 1 for all experiments, to encourage diversity in the generated items, and use a maximum token limit of 300. We explore the effect of varying other key hyperparameters:

- **Top P:** This parameter is based on nucleus sampling (Holtzman et al., 2019) and determines what fraction of log probabilities to consider when sampling, with larger values allowing more unlikely completions to be sampled. Prior work in LLM-based AIG has differed on this setting; some have used a value above 0.5 (Lee et al., 2023) and others a value at or below 0.5 (Liu et al., 2022). We therefore choose to experiment with both 0.5 and 1, as we theorized setting a higher value could lead to more diverse generations, but also increase the risk the items would lack construct validity.
- **Prompt Type:** We use a “simple” prompt following the structure shown in Figure 1. However, because the *AX* categories are highly specific, we reasoned that providing additional context about the categories may improve generation accuracy. We thus also experiment with “elaborated” prompts, which include additional information about each category, taken from the appendix on *AX*.<sup>5</sup>

We left all other hyperparameters at their defaults. We use the `text-davinci-003` endpoint,<sup>6</sup> and queried the API in December 2022. We generate 400 items, 100 for each category, and 25 for each hyperparameter combination (prompt type and top  $p$ ). We remove any duplicate items, items where the model did not generate a valid label, and items that match verbatim an item from *AX*.

Following best practices in scale development (Worthington and Whittaker, 2006) we conduct a content review on the generated items. Four Ph.D. students with prior publications in NLP, NLI, or psychometric AI were asked to rate the quality of the GPT-3 items. We ask our annotators to rate the relevance of the items for measuring the category, the clarity of the items (in terms of whether they have spelling or grammatical errors), whether the items have potentially harmful content, and their

<sup>5</sup><https://gluebenchmark.com/diagnostics>

<sup>6</sup>Prompts and generated items for reproducing our results are available on Github: <https://github.com/Advancing-Machine-Human-Reasoning-Lab/gpt3-item-generation/tree/main>

certainty in their annotations. Before beginning the study, we gave annotators detailed instructions they were asked to review in advance, including information about the *AX* categories, how to answer each of the ratings, and example ratings. We instructed annotators to rate items as “Completely irrelevant” if either the label was incorrect or the item did not match the target category. We followed standard practices in NLI research for determining what the correct label should be (Bowman et al., 2015), which all our annotators were informed of. In particular, annotators always assumed *p* and *h* referred to the same event or situation (Bowman et al., 2015). For determining category membership, we follow the definitions of each *AX* category provided by Wang et al. (2018), and developed a simple code book for determining this. The majority of the annotations were done synchronously in a four-hour annotation session. Per recommended practices for content analysis, each item was rated by every annotator (Putka et al., 2008). Annotators were encouraged to discuss items with each other and come to an agreement on what ratings should be used. Further details on the content review, including an example of the annotation interface, can be found in Appendix A.

For a generated item to pass the content review, we determined that all annotators must rate the item as very clear, either relevant or very relevant, that the item contained no harmful content, and that annotators were either sure or very sure of their predictions. Of the 400 items, 92 met these criteria across all categories, with at least 15 in every category passing. We sampled 15 at random from each category, balanced for the label, to obtain the GPT-3 generated items. In total, 60 items were sampled.

## 4 Experiments

We determined in Section 3 that GPT-3 can generate *AX* items that possess at least face validity evidence. But are these items really more valid and reliable measures of basic linguistic reasoning, given that we designed our prompts to induce this? To study this, we recruited human participants on Amazon Mechanical Turk<sup>7</sup> to complete both the GPT-3 items and the original human-written items.

102 participants residing in the United States, who had completed at least 50 HITs (human intelligence tasks) with an acceptance rate of at least 90%,

<sup>7</sup><https://www.mturk.com>

were recruited to take part in the study. We use the attention check items and quality control protocol from Laverghetta Jr. et al. (2021) to validate that our workers participated in good faith. Workers first completed an onboarding HIT where they were given five attention check items, whose format was identical to the *AX* items but by design, they were much easier to solve. This was meant to familiarize workers with the task and ensure they would likely give good response data. Workers who passed the onboarding then completed two more HITs, each containing half the GPT-3 items, and then two final HITs, each containing half the human-written items, and each of these HITs contained six attention checks spread evenly throughout the survey. Each worker’s submission was evaluated on every survey, and we followed the protocols developed by Laverghetta Jr. et al. (2021) to determine whether work should be accepted or rejected. Briefly, workers needed to get at least 60% accuracy on the survey, or at least 66% on the attention checks, and provide a *justification* for each response to show that they were truly paying attention. Further details on the protocol and payment structure for the human studies are included in Appendix B.

We ultimately gathered data from 18 participants and base the following analysis on this sample. Broadly, our goal is to compare the psychometric properties of the GPT-3 written items to the human-written items, focusing specifically on item difficulty, item discrimination, reliability (assessed using internal consistency), and convergent and discriminant validity. These are all important properties to analyze when establishing the validity and reliability of a new assessment (Bandalos, 2018), and when assessed using a measurement framework known as classical test theory (CTT), can be computed using small sample sizes. CTT essentially posits that an individual’s true proficiency on a cognitive task (their *true score*) can be decomposed into an observed (actual) score they obtain and an error term that represents the measurement error (Rust and Golombok, 2014). Note that this error is assumed to be random, and not systematic. Methods from CTT for assessing both validity and reliability are hence based on analysis of observed scores, and correlations between observed scores, where the observed scores are simply accuracy on the task:

$$\text{observed score} = \frac{\text{correct answers}}{\text{all answers}} \quad (2)$$

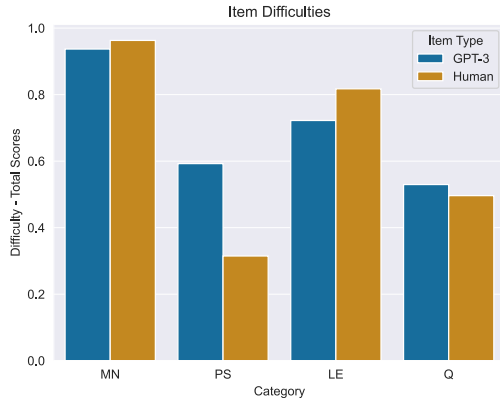


Figure 2: Mean item difficulties for each category, measured using total scores. Lower values indicate lower total scores, and hence more difficult items.

Although more sophisticated measurement theories have been developed (Embretson and Reise, 2013), they typically rely on latent variable modeling and require much larger sample sizes. Furthermore, in practice, establishing validity and reliability under CTT is often a first step in validating new assessments (Bandalos, 2018), which we believe justifies our focus on CTT in the present study.

#### 4.1 Analysis of Item Properties

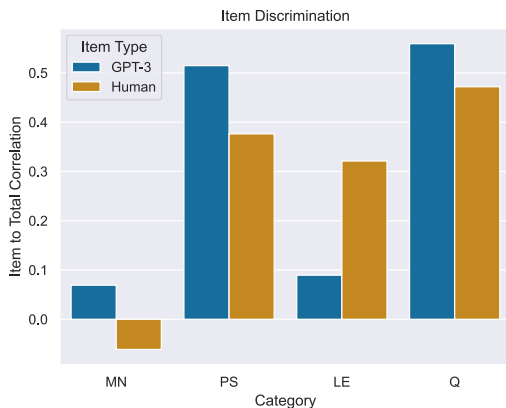


Figure 3: Mean item-to-total correlations for each category. Higher values indicate items are more predictive of a participant’s total score, and hence are more discriminating.

We begin by comparing mean item difficulties (Figure 2) and mean item discriminations (Figure 3) for both human and GPT-3 written items. Difficulty is based on the participants’ observed scores, and is equivalent to accuracy. Classical psychometrics dictates that items should have difficulties at approximately the midpoint between chance and

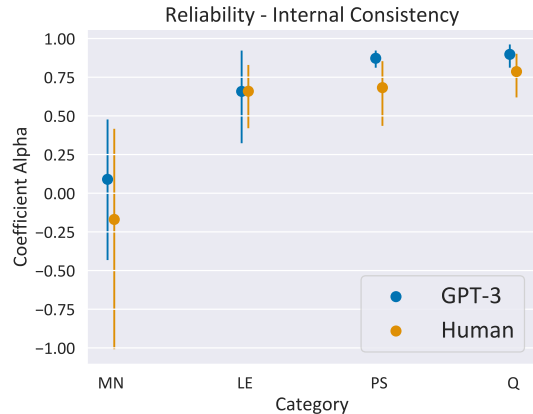


Figure 4: Coefficient  $\alpha$  for item responses in each category, comparing human-written to GPT-3 written items. Errors bars are 95% confidence intervals computed using Feldt’s method (Feldt et al., 1987). Higher values indicate better reliability and stronger validity evidence.

perfect scores (Lord, 1952), which in our case is roughly 70%. We again use item-to-total correlation to measure discrimination, and recall that item discrimination should be positive, with high values indicating better discrimination. We find that GPT-3 items are consistently closer to the optimal difficulty level than human-written items. GPT-3 items are also more discriminating than human-written ones, though a notable exception is for LE, where the GPT-3 items are noticeably less discriminating. As LE tests for all forms of lexical entailment, and is a much more broadly scoped construct than the others, lower discrimination is expected (Clark and Watson, 1995), though this does not fully explain the rather sizeable drop.

#### 4.2 Internal Consistency Reliability

Items on cognitive assessments should exhibit strong reliability, meaning that participants with similar ability levels should also respond in a similar fashion. A widely used measure of reliability is coefficient  $\alpha$  (Tavakol and Dennick, 2011), defined as:

$$\alpha = \frac{k}{k-1} \left( 1 - \frac{\sum_{i=1}^k \sigma_{y_i}^2}{\sigma_x^2} \right) \quad (3)$$

Where  $k$  is the total number of items,  $\sigma_x^2$  is the variance of total scores across all items, and  $\sigma_{y_i}^2$  is the variance of total scores for item  $i$ .  $\alpha$  ranges from  $-\infty$  to 1, and will be negative when there is greater within-subject variability than between-subject variability. Reliability should thus be maximized. We compute  $\alpha$  for both GPT-3 and human

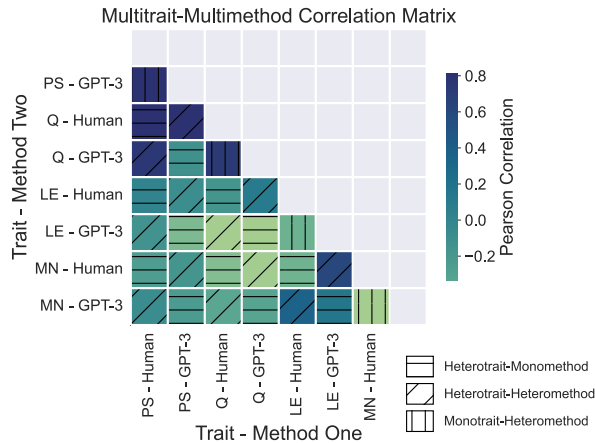


Figure 5: Results from the MTMM matrix, computed using Pearson correlations with total scores. Bluer colors indicate stronger correlation.

written items, doing so separately for each category, using the Pingouin Python library (Vallat, 2018). Reliabilities with 95% confidence intervals are shown in Figure 4. Across all categories, GPT-3 produces items with similar or better reliabilities compared to human-written items. MN is a special case, as  $\alpha$  for this category dips into the negative range, indicating poor validity evidence, though even in this case the GPT-3 items show much better reliability overall. Thus, the GPT-3 items appear to elicit more consistent responses among human participants.

### 4.3 Convergent and Discriminant Validity Evidence

The multi-trait multi-method (MTMM) matrix is a classic technique for evaluating the construct validity of measures and is often used when evaluating new instruments (Campbell and Fiske, 1959). The MTMM matrix shows the correlations between different cognitive constructs (the traits) when they are measured using different measurement techniques (the methods). In this framework, validity is defined in terms of the strength of the correlation between different trait / method combinations. In general, different methods should be strongly correlated when measuring the same trait (monotrait-heteromethod), and different traits measured using the same method should be weakly correlated (heterotrait-monomethod), per the definitions of convergent and discriminant validity (Campbell and Fiske, 1959).

We use this approach to evaluate the convergent and discriminant validity of the GPT-3 items. We

treat the *AX* category as the trait, and the method used to generate items (human written or generated by GPT-3) as the method and compute Pearson correlations between all possible combinations of trait and method, using the participant's total scores. Additionally, we check for significance using Bonferroni corrected p-values of 0.002.<sup>8</sup> Results are shown in Figure 5. Significant monotrait-heteromethod correlations were found for PS ( $\rho = 0.75$ ,  $p < 0.001$ ) but not for Q ( $\rho = 0.72$ ,  $p < 0.01$ ), MN ( $\rho = 0.06$ ,  $p < 0.5$ ) or LE ( $\rho = 0.20$ ,  $p < 0.5$ ). All heterotrait-monomethod correlations were insignificant ( $p > 0.1$ ), except for between PS and Q. For human-written items, the correlation was found to be significant ( $\rho = 0.81$ ,  $p < 0.001$ ), but not for GPT-3 written items ( $\rho = 0.16$ ,  $p < 0.5$ ). Collectively, these results indicate strong evidence for the discriminant validity of the GPT-3 items, given the lack of significant heterotrait-monomethod correlations. Evidence for convergent validity is strong for PS, and to a lesser extent Q,<sup>9</sup> but not for either MN or LE. Thus, the validity evidence for GPT-3 written items is just as strong, if not stronger, than for human-written items.

### 4.4 Analysis of Local Item Dependency

Recall that CTT assumes that measurement errors are due purely to random chance, and systematic error is not easily accounted for. One way this can be violated is from a phenomenon called local item dependence (LID). LID occurs between pairs of items, often whenever information needed to solve the items is interrelated. For example, LID is often a concern on reading comprehension assessments, because items that refer to the same text can inadvertently introduce local dependency on the common stimulus (Attali et al., 2022). Importantly, LID indicates that errors on items are interrelated in a way other than proficiency on the construct, and hence imply systematic error in the measurement.

As Attali et al. (2022) notes, LID is an even greater concern in the context of AIG, as GPT-3 may have generated items in a programmatic and somewhat redundant fashion. Perhaps as an artifact of how *AX* was constructed, we also found many human-written items had highly similar linguistic structures, which we reasoned could cause GPT-

<sup>8</sup>Rounded to three decimal places.

<sup>9</sup>The monotrait-heteromethod correlations for Q were strong, even though they did not meet the Bonferroni-corrected significance level.

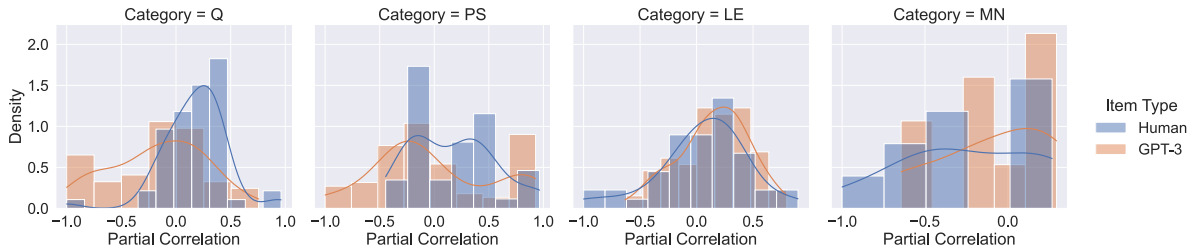


Figure 6: Density plots (computed using kernel density estimation) of partial Pearson correlations computed for each category, controlling for the participants’ total scores per category. Item pairs where one or both items have 0 variance are excluded. Partial correlations greater than 0.3 indicate LID, and distributions which peak closer to 0 have fewer item pairs with LID.

3 to generate items based on a common stimulus, which might inadvertently introduce LID. We thus follow Attali et al.’s protocol and, for each category and for both the human-written and GPT-3 written items, we compute the partial correlations between all pairs of items in each category, controlling for total scores. Following prior work (Christensen et al., 2017; Attali et al., 2022), we use a threshold of 0.3 correlation or higher as indicating LID, and we plot the density distributions of the partial correlations in each category. Results are shown in Figure 6. We find that, even with the human-written items, LID appears to be present in all categories except for MN, though even in this case we observe strong anti-correlations. It does not appear, however, that the GPT-3 items have made LID significantly worse. Distributions are often similar between the item types, and in some cases, GPT-3 distributions appear closer to zero, indicating fewer pairs with LID. We thus surmise that LID is no greater a concern for GPT-3 written items than it was for human-written items.

#### 4.5 Scaling Up to GPT-4

OpenAI’s most recent LLM, GPT-4,<sup>10</sup> was released after the completion of our testing of the GPT-3 items. Given the large gains in performance reported for GPT-4 across myriad tasks, we chose to perform preliminary analysis on the quality of items generated by GPT-4, this time running only the content review.<sup>11</sup> We use the same content experts and follow an identical protocol for the review. We chose not to generate items for MN, due to the very poor validity evidence for items in this category. Hyperparameters and prompts remain the

<sup>10</sup><https://openai.com/research/gpt-4>

<sup>11</sup>Due to time constraints, we could not run a more detailed analysis on the GPT-4 written items, and leave this to future work.

same, and we use the `gpt-4` endpoint in the API. To keep results as comparable as possible across models, we chose not to use the system context or other chat features provided for GPT-4, and instead administer the prompts in a single shot. We generate 18 items per category, totaling 54 across the three categories tested. After running deduplication and dropping items with invalid labels, we administer the remaining items to our content experts. We were specifically interested in whether our experts would report the GPT-4 items as being any more relevant for measuring the target construct as compared to GPT-3. We graph the annotator distributions for PS in Figure 7, and show results for LE and Q in Appendix C. Surprisingly, we find results from GPT-4 to be mixed. Although GPT-4 generates a larger fraction of items labeled as either “Relevant” or “Very relevant” for Q, it generates fewer such items for LE and PS. As GPT-4 is designed to function more like a chatbot than GPT-3, it is possible our prompts need to be restructured to make better use of the model’s capabilities, but more experiments are needed to explore this.

## 5 Discussion and Conclusion

Collectively, our results demonstrate that LLMs can generate items with superior validity evidence, even for constructs that have undergone limited psychometric analysis. GPT-3 items were found to have better discrimination and reliability, while maintaining strong convergent, discriminant, and content validity. LID, while confirmed to be present in both item types, appeared no worse and perhaps slightly better in GPT-3 items. These positive results, while clearly present for PS and Q, were less clear for MN and LE, and validity evidence as a whole appeared strongest for the categories testing the most narrowly scoped constructs.

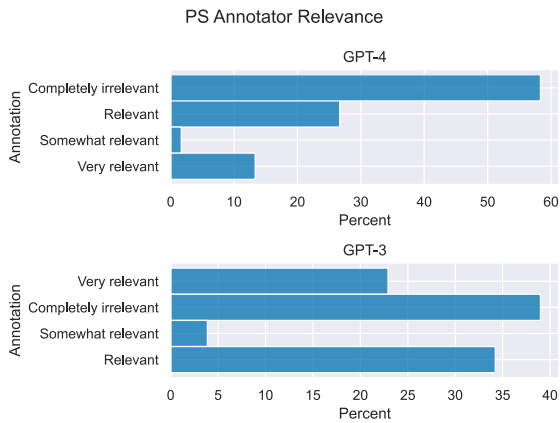


Figure 7: Distribution of annotator relevance scores (checking that the item both has a correct label and matches the category) for both GPT-3 and GPT-4 items, on items from the PS category. A lower percentage of items marked as “Completely irrelevant” indicates stronger evidence of the content validity of items generated using that model.

Though promising, our results come with limitations that should be addressed in future work. The small sample size we collected makes it difficult to assess the generalizability of our findings. This also prevented us from running any analysis of internal structure or differential item functioning (DIF) using methods from factor analysis or item response theory, as these models require large sample sizes (Min and Aryadoust, 2021). As items generated by GPT-3 should contain no DIF and have similar factor structures as items written by humans, these are important analyses to explore in future work. We also did not examine the *diversity* of the generated items, in other words, how thoroughly the model explored the construct space. It is a well-known problem in psychometrics that having too many similarly worded items can inflate the reliability and reduce the validity of a measure (Clark and Watson, 1995), and our results may have been susceptible to this. A related problem is ensuring that the distribution of labels in the generated items remains balanced, and while we took steps to account for this, we did find that the distribution of GPT-3 items was somewhat unbalanced. For example, there were far fewer neutral items than either entailment or contradiction. Improving the prompt design to account for diversity and other psychometric properties simultaneously is a fruitful direction for future work. Our experiment with GPT-4, while disappointing, was also quite limited and should be expanded upon. We deliberately kept the prompt

design as similar as possible between the two models, to avoid possible confounds. Making effective use of the system query and changing the structure of the prompts to suit a conversational style could lead to much better results, however. Finally, although we believe NLI is a good task to use for initial experimentation, we also acknowledge that it is significantly different from the tasks of interest in education (e.g., question answering), and future work should explore our approach on tasks with stronger educational applications.

LLMs have the potential to greatly ease the burden of scale development, and transform educational and psychological measurement. Our results contribute to the growing field of LLM-based automated item generation, and demonstrate the potential these methods have for generating valid and reliable items at a scale that would have previously been impossible. Further research, combining our approach with more advanced prompting strategies, or zero-shot parameter estimation, could conceivably lead to a system that generates high-quality items in a fully autonomous fashion, which would transform the practice of writing and validating test items.

## Limitations

We emphasize that our research is exploratory and the generated items we produced should not be used for making critical evaluations of cognitive skillsets in either humans or LLMs. As discussed in Section 5, our small sample size makes it difficult to draw broad conclusions about the generalizability of our findings, and practical considerations regarding the annotation study limited our ability to thoroughly explore the prompt space. While we chose GPT-3 due to its ease of use and the fact that most psychometricians would likely be aware of it, we also acknowledge that OpenAI has released few details on how this model is trained or updated, which hampers the reproducibility of our results. We also acknowledge that more recent OpenAI LLMs, including ChatGPT and GPT-4, have been released since this work is completed, and that our preliminary experiments using GPT-4 do not give us a full understanding of the capabilities of this model. However, given that we were still able to perform detailed experiments using the GPT-3 items, and these items proved to have superior validity evidence across multiple trials, we do not believe the existence of more recent LLMs negates

our results. Finally, it is also well known that LLMs can produce biased, toxic, or other forms of harmful text content (Liang et al., 2021). While we took steps to account for this in our content review, future work must keep this possibility in mind and carefully analyze generated items for potentially harmful content. A related problem is the risk of GPT-3 items propagating disadvantages against historically marginalized groups. For example, the items may have relied on cultural context or other information that would give an unfair advantage to certain populations. Given that we lacked a sufficient sample size and did not collect personally identifiable information from participants, we could not run DIF analysis to check for this, and cannot state definitively that DIF is not present.

## Acknowledgements

We would like to thank Logan Fields, Animesh Nighojkar, and Zaid Marji for assisting us with the content review. Part of this research was sponsored by the DEVCOM Analysis Center and was accomplished under Cooperative Agreement Number W911NF-22-2-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

## References

- Ahmed M Asfahani. 2022. The impact of artificial intelligence on industrial-organizational psychology: A systematic review. *The Journal of Behavioral Science*, 17(3):125–139.
- Yigal Attali, Andrew Runge, Geoffrey T. LaFlair, Kevin Yancey, Sarah Goodwin, Yena Park, and Alina A. von Davier. 2022. [The interactive reading task: Transformer-based automatic item generation](#). *Frontiers in Artificial Intelligence*, 5.
- Deborah L Bandalos. 2018. *Measurement theory and applications for the social sciences*. Guilford Publications.
- Isaac I Bejar, René R Lawless, Mary E Morley, Michael E Wagner, Randy E Bennett, and Javier Revuelta. 2002. A feasibility study of on-the-fly item generation in adaptive testing. *ETS Research Report Series*, 2002(2):i–44.
- Samuel 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.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Donald T Campbell and Donald W Fiske. 1959. Convergent and discriminant validation by the multitrait-multimethod matrix. *Psychological bulletin*, 56(2):81.
- Sophia Chan, Swapna Somasundaran, Debanjan Ghosh, and Mengxuan Zhao. 2022. Agree: A system for generating automated grammar reading exercises. *arXiv preprint arXiv:2210.16302*.
- Xieling Chen, Haoran Xie, Di Zou, and Gwo-Jen Hwang. 2020. Application and theory gaps during the rise of artificial intelligence in education. *Computers and Education: Artificial Intelligence*, 1:100002.
- Song Cheng, Qi Liu, Enhong Chen, Zai Huang, Zhenya Huang, Yiyi Chen, Haiping Ma, and Guoping Hu. 2019. Dirt: Deep learning enhanced item response theory for cognitive diagnosis. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 2397–2400.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.
- Karl Bang Christensen, Guido Makransky, and Mike Horton. 2017. Critical values for yen’s q 3: Identification of local dependence in the rasch model using residual correlations. *Applied psychological measurement*, 41(3):178–194.
- Lee Anna Clark and David Watson. 1995. Constructing validity: Basic issues in objective scale development. *Psychological Assessment*, 7(3):309.
- Robin Cooper, Dick Crouch, Jan Van Eijck, Chris Fox, Johan Van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, et al. 1996. Using the framework. Technical report, Technical Report LRE 62-051 D-16, The FraCaS Consortium.
- Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment: First*

- PASCAL Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers*, pages 177–190. Springer.
- 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.
- 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, Copenhagen, Denmark. Association for Computational Linguistics.
- Susan E Embretson and Steven P Reise. 2013. *Item response theory*. Psychology Press.
- Leonard S Feldt, David J Woodruff, and Fathi A Salih. 1987. Statistical inference for coefficient alpha. *Applied psychological measurement*, 11(1):93–103.
- Friedrich M Götz, Rakoen Maertens, Sahil Loomba, and Sander van der Linden. 2023. Let the algorithm speak: How to use neural networks for automatic item generation in psychological scale development. *Psychological Methods*.
- Ivan Hernandez and Weiwen Nie. 2022. The ai-ip: Minimizing the guesswork of personality scale item development through artificial intelligence. *Personnel Psychology*.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.
- Antonio Laverghetta Jr., Animesh Nigohjkar, Jamshidbek Mirzakhlov, and John Licato. 2021. [Can transformer language models predict psychometric properties?](#) In *Proceedings of \*SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 12–25, Online. Association for Computational Linguistics.
- Philseok Lee, Shea Fyffe, Mina Son, Zihao Jia, and Ziyu Yao. 2023. [A paradigm shift from “human writing” to “machine generation” in personality test development: an application of state-of-the-art natural language processing](#). *Journal of Business and Psychology*, 38:163–190.
- Paul Pu Liang, Chiyu Wu, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2021. [Towards understanding and mitigating social biases in language models](#). In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6565–6576. PMLR.
- Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022. [WANLI: Worker and AI collaboration for natural language inference dataset creation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6826–6847, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Frederic M Lord. 1952. The relation of the reliability of multiple-choice tests to the distribution of item difficulties. *Psychometrika*, 17(2):181–194.
- Rakoen Maertens, Friedrich Götz, Claudia R Schneider, Jon Roozenbeek, John R Kerr, Stefan Stieger, William Patrick McClanahan III, Karly Drabot, and Sander van der Linden. 2021. The misinformation susceptibility test (mist): A psychometrically validated measure of news veracity discernment.
- Shangchao Min and Vahid Aryadoust. 2021. A systematic review of item response theory in language assessment: Implications for the dimensionality of language ability. *Studies in Educational Evaluation*, 68:100963.
- Elham Mousavinasab, Nahid Zarifsanaiey, Sharareh R. Niakan Kalhori, Mahnaz Rakhshan, Leila Keikha, and Marjan Ghazi Saeedi. 2021. Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods. *Interactive Learning Environments*, 29(1):142–163.
- Yixin Nie, Xiang Zhou, and Mohit Bansal. 2020. [What can we learn from collective human opinions on natural language inference data?](#) In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9131–9143, Online. Association for Computational Linguistics.
- Animesh Nigohjkar and John Licato. 2021. [Improving paraphrase detection with the adversarial paraphrasing task](#). 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 7106–7116, Online. Association for Computational Linguistics.
- Simon Ott, Adriano Barbosa-Silva, Kathrin Blagec, Jan Brauner, and Matthias Samwald. 2022. Mapping global dynamics of benchmark creation and saturation in artificial intelligence. *Nature Communications*, 13(1):6793.
- Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. *Advances in neural information processing systems*, 28.
- Septian Eko Prasetyo, Teguh Bharata Adji, and Indriana Hidayah. 2020. [Automated item generation: Model and development technique](#). pages 64–69. IEEE.



- Dan J Putka, Huy Le, Rodney A McCloy, and Tirso Diaz. 2008. Ill-structured measurement designs in organizational research: Implications for estimating interrater reliability. *Journal of Applied Psychology*, 93(5):959.
- Alec Radford, Jeffrey 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. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Manav Rathod, Tony Tu, and Katherine Stasaski. 2022. Educational multi-question generation for reading comprehension. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 216–223, Seattle, Washington. Association for Computational Linguistics.
- Cristobal Romero and Sebastian Ventura. 2020. Educational data mining and learning analytics: An updated survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(3):e1355.
- John Rust and Susan Golombok. 2014. *Modern psychometrics: The science of psychological assessment*. Routledge.
- 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.
- Stephen Stark, Oleksandr S Chernyshenko, and Nigel Guenole. 2011. Can subject matter experts’ ratings of statement extremity be used to streamline the development of unidimensional pairwise preference scales? *Organizational Research Methods*, 14(2):256–278.
- Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.
- Mohsen Tavakol and Reg Dennick. 2011. Making sense of cronbach’s alpha. *International journal of medical education*, 2:53.
- Mikke Tavast, Anton Kunnari, and Perttu Hämäläinen. 2022. Language models can generate human-like self-reports of emotion. In *27th International Conference on Intelligent User Interfaces*, pages 69–72.
- Raphael Vallat. 2018. Pingouin: statistics in python. *J. Open Source Softw.*, 3(31):1026.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Matthias von Davier. 2018. Automated item generation with recurrent neural networks. *Psychometrika*, 83:847–857.
- Reece Walsh, Mohamed H Abdelpakey, Mohamed S Shehata, and Mostafa M Mohamed. 2022. Automated human cell classification in sparse datasets using few-shot learning. *Scientific Reports*, 12(1):2924.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.
- 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, New Orleans, Louisiana. Association for Computational Linguistics.
- Roger L Worthington and Tiffany A Whittaker. 2006. Scale development research: A content analysis and recommendations for best practices. *The counseling psychologist*, 34(6):806–838.
- Bowei Zou, Pengfei Li, Liangming Pan, and Ai Ti Aw. 2022. Automatic true/false question generation for educational purpose. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 61–70, Seattle, Washington. Association for Computational Linguistics.
- Jiyun Zu, Ikkyu Choi, and Jianguang Hao. 2023. Automated distractor generation for fill-in-the-blank items using a prompt-based learning approach. *Psychological Testing and Assessment Modeling*, 65(2):55–75.

## A Details on Content Review

Content review ratings were collected via Qualtrics.<sup>12</sup> We developed five items to ask our experts:

<sup>12</sup><https://www.qualtrics.com>

QID8. **Premise:** Sarah went for a run in the park.  
**Hypothesis:** Sarah went for a walk in the park.  
**Label:** neutral  
**Construct (Category):** Lexical Entailment  
 Prompt ID: style\_1\_le  
 Top P: 0.5

See below  
○

QID9. How relevant is the item for measuring the NLI construct?

Completely irrelevant ○	Somewhat relevant ○	Relevant ○	Very relevant ○
----------------------------	------------------------	---------------	--------------------

QID10. How clear is the wording of the item (e.g., are there syntax errors, odd word choice, etc.)?

Not clear, major revisions ○	Somewhat clear, some revisions ○	Clear, slight revisions ○	Very clear, no revisions ○
---------------------------------	-------------------------------------	------------------------------	-------------------------------

QID11. Does the item contain potentially harmful content?

Yes	No
-----	----

Figure 8: The annotation interface for the content review.

1. **Item Relevance:** This question concerned the usefulness of the item for measuring the construct. Experts could rate items as “Completely irrelevant”, “Somewhat relevant”, “Relevant”, or “Very relevant”. At a basic level, items needed have both a correct label and test for the target category. If *either* of these were false, experts were instructed to rate the item as “Completely irrelevant”. Experts were instructed to rate items as “Somewhat relevant” if the prior checks passed, but knowledge of the category was not critical to solving the item. An example of this would be an item from MN where the negated clause does not change at all from *p* to *h*. If knowledge of the category was critical, and all prior checks passed, experts were instructed to rate the item as “Relevant”. “Very relevant” was reserved for items that experts judged as being highly discriminating, which we included based on prior work demonstrating experts can effectively evaluate latent properties of items (Stark et al., 2011). We left the exact judgment of what constituted a highly discriminating item up to the discretion of the ex-

perts, and we encouraged them to discuss this and reach an agreement for each item deemed “Very relevant”.

2. **Item Clarity:** This question concerned how clear the wording of the item is, and whether it contains spelling or grammatical errors. Experts could rate items as “Not clear, major revisions”, “Somewhat clear, some revisions”, “Clear, slight revisions”, and “Very clear, no revisions”. “Not clear, major revisions” was reserved for cases where items contained any spelling or grammatical errors. This also included cases with unterminated punctuation (e.g., an opening ‘(’ that was not closed). Both “Somewhat clear, some revisions” and “Clear, slight revisions” were reserved for cases where the prose of the item was unorthodox (e.g., GPT-3 generated an odd word choice or an unusual phrase). Experts were instructed to rate “Very clear, no revisions” if items were both grammatically correct and contained no unusual wording that made the item needlessly difficult to understand.

3. **Potentially Harmful Content:** This was included to ensure that GPT-3 did not generate offensive or otherwise harmful content in the items, though we did not expect this to be an issue in general as *AX* items were written in a fairly neutral tone and avoided covering controversial social issues or explicitly targeting identified subgroups. Experts were instructed to check if the items contained any content related to race, ethnicity, religion, or other identifiable characteristics that might be considered offensive to members of those groups. Importantly, *AX* does contain items related to U.S. politics circa 2018 that we reasoned might lead to toxic generations regarding political ideology. We made experts aware of this but instructed them to *only* rate such items as harmful if the content explicitly attacked a political ideology or its adherents. There were only two options for this item, “yes” or “no”.
4. **Annotator Certainty:** Finally, using a four-point Likert scale, we asked annotators to rate how sure they were of their ratings.

Figure 8 shows the annotation interface. Experts were given the full item content and the label generated by GPT-3, as well as additional data about the hyperparameters used which they did not need to refer to. They were free to move back and forth within the survey and revise their responses later if they wished. Most annotations were completed in a synchronous session, and all annotators began their work in this session to ensure the task instructions were clear and to train them on how to rate each item. Importantly, we did not ask raters to edit any item content to improve its quality, as we were interested in the quality of GPT-3 written items without human intervention.

For determining category membership, we developed a codebook based on the presence of certain keywords in the item content, and either  $p$  or  $h$  needed to contain at least one of these keywords to pass content validity. For example, for *Q*, either  $p$  or  $h$  needed to contain either a universal (all, none) or existential (some, many, most, etc.) quantifier in natural language to pass. We developed an initial list of keywords based on both the appendix covering *AX* in Wang et al. (2018), and by manually inspecting the items in each category to locate additional keywords. During the content review, experts could also suggest additional keywords,

and if all annotators agreed, these new keywords were added to the codebook. Table 2 show all the keywords used across categories. LE was the only category that did not follow this protocol for determining category membership. As LE tests for all forms of entailment at the word level, there is no predetermined list of keywords that can be used to determine LE membership. Therefore, for LE, we used the rule that  $p$  and  $h$  must differ by only one word, with the only exception being if other words needed to be changed to keep the sentences grammatically correct.

## B Details on Human Study

We follow many of the same protocols from Laverghetta Jr. et al. (2021) for conducting our human study. In particular, they employed attention check NLI items taken from the ChaosNLI dataset (Nie et al., 2020), which collected 100 human ratings to a subset of SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2018) items. Only items which at least 90% of the workers agreed on the correct label were used, and hence they are presumably quite easy to answer correctly. In addition, Laverghetta Jr. et al. (2021) also asked workers to justify their response to each item, which was used as an additional check to ensure workers were paying attention during the task. We follow their protocol and check that workers do not copy text from the item as their justification, that the justification is not used multiple times, that it is clearly related to the item content,<sup>13</sup> and that the justification is not a nonsensical word or phrase (e.g. “good” or “nice question”). Collectively, the following quality control procedure was used for each survey:

1. Submissions with duplicate IP addresses or worker IDs were dropped.
2. Submissions with less than 40% accuracy, or less than 60% with less than 66% on attention checks, were dropped.
3. Submissions whose justifications did not meet the above criteria were also dropped.

All other submissions were accepted, and at each stage passing workers were given qualifications to proceed to the next survey. If however, workers

<sup>13</sup>In some instances, workers appeared to copy text from external websites that was completely unrelated to the question.

Category	Keywords
MN	un-, non- ir-, dis-, im-, il-, in-, -n't, not, never, no
PS	un-, non- ir-, dis-, im-, il-, in-, -n't, not, no, and, or, if
Q	all, no, some, many, most, none, every, several, each, one other, only, nearly all, the , part of

Table 2: Keywords used to determine category membership. Leading and trailing “-” indicate suffixes and prefixes, respectively.

failed a given stage, they were not allowed to proceed. In total, we administered five separate HITs and used Qualtrics to gather all responses. Workers were paid \$8.00 for each HIT, except for the initial onboarding HIT, where they were paid \$0.10,<sup>14</sup> and had one hour to complete each HIT. Workers were told they would be compensated for each survey completed successfully, to encourage consistently high-quality work. Workers gave informed consent to participate prior to beginning each HIT, and could withdraw at any time. Workers could appeal any rejections made, however, we also clearly stated submissions would be checked for quality control purposes, and may be dropped if evidence of bad-faith responses was found. All work was done anonymously; workers were not asked to provide us with any personally identifiable information at any stage.

Finally, we also considered extending Laverghetta Jr. et al.’s protocol to check for AI-generated text for the explanations, in case workers attempted to use ChatGPT or another LLM during the survey. We examined several detectors for AI-written text, including one developed by OpenAI.<sup>15</sup> However, we found that currently available models require too much text to be helpful for our study. Participants were asked to only briefly explain their thought process with at most one sentence, which was far too short for current detectors to make a classification. Therefore, we did not include any check for AI-generated text, but we strongly encourage future work to consider this and investigate other possible safeguards against workers cheating on the task using LLMs.

### C Additional Results from GPT-4

Figures 9 and 10 compare the annotator relevance scores between GPT-3 and GPT-4 items, for LE and Q.

<sup>14</sup>This HIT contained only 5 items and was meant to be finished quickly.

<sup>15</sup><https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text>

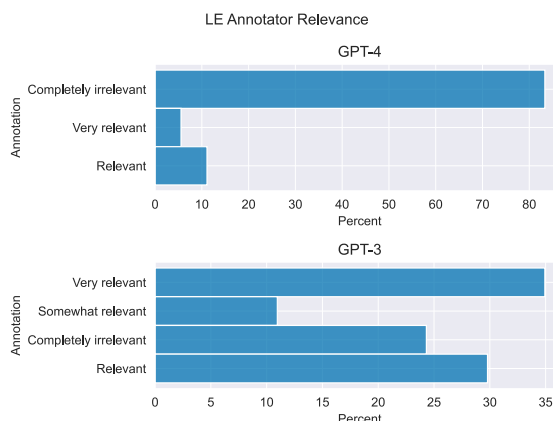


Figure 9: Distribution of annotator relevance scores for LE.

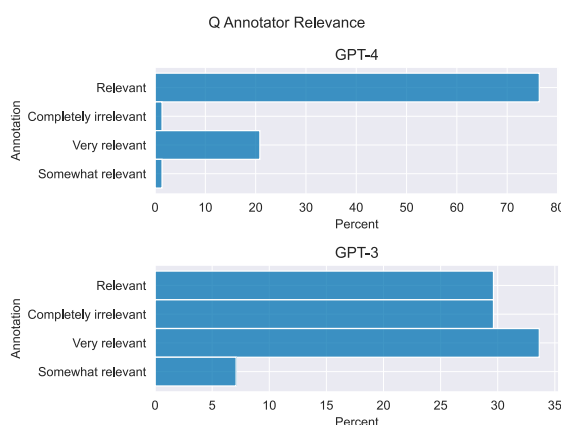


Figure 10: Distribution of annotator relevance scores for Q.

# Span Identification of Epistemic Stance-Taking in Academic Written English

Masaki Eguchi and Kristopher Kyle

Learner Corpus Research and Applied Data Science Lab

<https://lcr-ads-lab.github.io/LCR-ADS-Home/>

Department of Linguistics, University of Oregon

{masakie, kkyale2}@uoregon.edu

## Abstract

Responding to the increasing need for automated writing evaluation (AWE) systems to assess language use beyond lexis and grammar (Burstein et al., 2016), we introduce a new approach to identify rhetorical features of stance in academic English writing. Drawing on the discourse-analytic framework of engagement in the Appraisal analysis (Martin & White, 2005), we manually annotated 4,688 sentences (126,411 tokens) for eight rhetorical stance categories (e.g., PROCLAIM, CONTRIBUTION) and additional discourse elements. We then report an experiment to train machine learning models to identify and categorize the spans of these stance expressions. The best-performing model (RoBERTa + LSTM) achieved macro-averaged F1 of .7208 in the span identification of stance-taking expressions, slightly outperforming the intercoder reliability estimates before adjudication (F1 = .6629).

## 1 Introduction

Automated writing evaluation (AWE) systems make it possible to assess students' writings and provide useful feedback efficiently (Shermis & Burstein, 2013). From the language assessment perspective, however, *usefulness* is multifaceted (e.g., Bachman & Palmer, 1996) and, in many parts, depends on what areas of writing ability a given system can measure and give feedback on (Huawei & Aryadoust, 2023). While many AWE systems to date focus on lexical, syntactic, organizational, and topical aspects of students' writing (e.g., Attali, 2007), the construct of writing (i.e., writing skill) is known to be far more complex and includes pragmatic and rhetorical knowledge



Figure 1: A sample output of the best-performing system reported in this study. The excerpt was taken from the ICNALE corpus (Ishikawa, 2013).

(Bachman & Palmer, 2010; Sparks et al., 2014). Accordingly, recent studies have included constructs such as discourse moves and steps (e.g., Cotos, 2014), source use and citations (Burstein et al., 2018; Kyle, 2020), and argument structures using Rhetorical Structure Theory (Fiocco et al., 2022). Given the increasing focus on the assessment of the ability to construct effective persuasive texts (Sparks et al., 2014), innovative use of NLP is needed how to assess these social and rhetorical constructs of writing (e.g., Burstein et al., 2016; Carr, 2013; Lu, 2021).

One area that has received relatively little attention in the literature on AWE is the notion of evaluative language or stance-taking (Biber & Finegan, 1988; Hunston & Thompson, 2000; Xie, 2020). In the computational linguistics context, the notion of stance is often discussed in relation to the stance-detection task, where the objective is to categorize whether a text producer is *in favor*, *against*, or *None*, toward a certain topic (e.g., Schiller et al., 2021). However, from the language assessment perspective, researchers are more interested in the rhetorical strategies used to express a nuanced stance instead of the binary classification of positions (Biber, 2006; Biber & Finegan, 1988; Hyland, 2005). In applied linguistics research, evaluative language essentially concerns how writers express their stance on a topic of discussion or express their emotions or feelings on an entity (see Xie, 2020).

This paper reports the development and empirical evaluation of an end-to-end system to identify and categorize epistemic evaluative meanings in academic written discourse (see Figure 1 for illustration). We specifically draw on the discourse-analytic framework of the engagement system in the appraisal analysis (Martin & White, 2005) to create a gold-standard corpus of academic English. We then train an end-to-end span identification systems that can undertake stance analysis under the discourse functional framework. The free online demo of the current span identification system is accessible through Hugging Face Space<sup>1</sup>.

## 2 Background

### 2.1 Evaluative language

English for Academic Purposes (EAP) research often investigates evaluative language through corpus-based or discourse-analytic methods (Xie, 2020). Both approaches have both benefits and drawbacks. Qualitative discourse analysis allows researchers to analyze nuanced stance-taking strategies using contextual information; however, this limits the scalability of the analysis and thus cannot be used for large-scale standardized testing situations. Corpus-based approaches (e.g., Bax et al., 2019; Biber, 2006; Yoon, 2017) can overcome the issue of scalability. However, most tools rely

extensively on lexical and syntactic features (e.g., dictionary lookups of relevant vocabulary filtered for particular POS tags). Accordingly, these corpus approaches tend to neglect the fact that evaluative language can be poly-functional depending on the surrounding context. For example, very few corpus tools disambiguate whether the verb *suggest* is used to attribute an idea to external sources (*The authors suggest that ...*) or to hedge the writers' own view (e.g., *We suggest that ...*). Therefore, a probabilistic approach to identify the function in which the evaluative language is used is necessary to overcome the dilemma faced in the two approaches.

### 2.2 The engagement system

In this study, we draw on the framework of *engagement* in the appraisal analysis (Martin & White, 2005; White, 2003) as a theoretical framework for annotating functional categories of stance-taking expressions. According to Martin and White (2005), engagement concerns “locutions which provide the means for the authorial voice to position itself with respect to, and hence to ‘engage’ with, the other voices and alternative positions construed as being in play in the current communicative context” (p.94). In this discourse-analytic framework, parts of sentences (or clauses) are classified into different stances writers take. For example, a writer can present his/her idea as if it is a fact (e.g., *The banks have been greedy*; Martin & White, 2005). The use of present tense in the example implies that the statement does not recognize potential alternative realities and is thus termed **MONOGLOSS** by Martin & White (2005). Alternatively, a writer can display their awareness of other positions on the topic of discussion, using various heteroglossic strategies. These include, for example, **ATTRIBUTE** (e.g., *I heard on the recent news that the banks have been greedy*), **COUNTER** (e.g., *Although you might disagree, the banks have been greedy*), and **CONCUR** (e.g., *Everyone agrees that the banks are greedy*), etc. (see a complete list of discourse choices in Section 3.4).

The engagement system has been shown useful in describing nuanced ways in which writers position themselves against possible alternative views, for example, in peer-reviewed academic

---

<sup>1</sup><https://huggingface.co/spaces/egumasa/engagement-analyzer-demo>

paper (e.g., Chang & Schleppegrell, 2011; X. Xu & Nesi, 2017), university written assignments (e.g., Lancaster, 2014; Wu, 2007), and second language writing research (e.g., Lam & Crosthwaite, 2018). However, the analysis requires intensive manual coding because of the lack of automated tools that classifies the discourse-semantic category of engagement reliably. This means that in its current state, the engagement system cannot be applied to any large-scale educational applications. To benefit from the theoretical insights of discourse analysis in educational practices, this methodological obstacle needs to be overcome. The current study attempts to fill this gap using a supervised machine-learning approach.

### 2.3 Span identification

In this study, the task of identifying the evaluative language of engagement is conceptualized as a span identification task (see Gu et al., 2022; Papay et al., 2020). Span identification is a task of identifying boundaries of expressions in the input text and assigning a label (discourse-semantic one in the current study). Span identification has been used for a range of applications, including entity extraction (Gu et al., 2022), quoted material detection (Pareti, 2016), and toxic word detection (Rao, 2022). Particularly the latter two tasks are directly relevant to the current task because it attempts to identify text segments that may not be easily determined by particular grammatical features (e.g., noun chunks).

Recent span identification architectures (e.g., Gu et al., 2022; Rao, 2022) leverage large encoder-based pre-trained Transformer models (Devlin et al., 2019; Liu et al., 2019). For example, Gu et al. (2022) compared three approaches to formulate span identification tasks—Sequence Tagging, Span Enumeration, and Boundary Prediction. According to Gu et al. (2022), tagging is similar to NER in that each token is predicted under the BIO scheme (e.g., Papay et al., 2020). Span enumeration approaches the task by considering all spans within specified  $n$  lengths as candidates (as in Lee et al., 2017). Finally, boundary prediction takes a supervised approach to predict the start and end of spans. In the latter two approaches, span representations are created by pooling a set of token embeddings within the candidate spans (e.g., start and end tokens) (see Fu et al., 2021; Gu et al., 2022). Using the RoBERTa-base (Liu et al., 2019) and T5-base encoder (Raffel

et al., 2020), Gu et al. (2022) concluded that while the three had relative (dis)advantages, recall-focused tasks may benefit from span enumeration and boundary prediction.

In previous span identification architectures, researchers have often used additional contextualization by adding an additional Bi-LSTM layer on top of the transformer embeddings. However, the results appear mixed depending on the nature of the task and dataset (Gu et al., 2022; Papay et al., 2020). Therefore, a secondary goal of this study is to test whether we observe the benefits of additional contextual information via additional Bi-LSTM when the task does appear to require fine-tuned contextual information due to the discourse oriented nature of the proposed task (see Sections 2.1 and 2.2; see examples of the verb *suggest*).

### 2.4 Contribution of this study

The main contributions of this paper are two-fold. First, we present a new annotation scheme of academic English writing drawing on the discourse-analytic framework of the engagement (Martin & White, 2005) and present annotated dataset using the developed scheme (Section 3). Second, we present a new end-to-end model that can identify and categorize the span of engagement strategies (see Figure 1).

## 3 Engagement Discourse Treebank (EDT)

The EDT currently comprises 4,688 sentences with manually annotated engagement resource spans (126,411 tokens; 11,856 spans), which were sampled from corpora of academic English or closely related genres (see definition of in-domain text below). The version of EDT used to train the machine learning models presented in this paper is accessible at <https://github.com/LCR-ADS-Lab/Engagement-Discourse-Treebank>. The most recent version of the annotation guideline is accessible through the following GitHub page: <https://egumasa.github.io/engagement-annotation-project/>.

### 3.1 Definition of in-domain text

When developing a new dataset for an NLP task, it is important to clearly define the domain of texts to sample the annotation data to ensure the correspondence between the gold-standard

annotation and the kind of data to make inferences (Ramponi & Plank, 2020). A precise definition of in-domain text is also important from the AWE perspective since the degree of correspondence will influence the degree to which the AWE is able to assess the language use in the Target Language Use domain in language assessment (TLU domain; Bachman & Palmer, 2010). Following these two related concepts, we defined the in-domain text of EDT as academic written English of various genres written by both first- and second-language writers of English.

### 3.2 Source corpora

Annotation data was widely sampled from pre-existing corpora to represent the in-domain texts (see section 3.1 for definition). A major portion of data was sampled from two corpora of university written assignments—the British Academic Written English (Alsop & Nesi, 2009) and the Michigan Corpus of Upper-level Student Papers (Römer & O'Donnell, 2011)—representing first- and second-language writers of English. The remaining portion of data was sampled from a combination of corpora documenting timed essays by second-language writers with various backgrounds and proficiency levels (Blanchard et al., 2013; Ishikawa, 2013; Yannakoudakis et al., 2011). The selection of a wide range of sources, instead of commonly used data sources, such as Wall Street Journal articles, allowed us to represent the characteristics of in-domain texts.

### 3.3 Minimal context approach

During the corpus sampling, we opted for a minimal context window strategy (i.e., three-sentence) to achieve a compromise between the validity of the annotation and any practical considerations (e.g., budget, time constraints, copyrights of source corpora). In an ideal situation, the unit of analysis for annotation should be the entire document, particularly because the object of the annotation is discourse semantics; however, there are arguably advantages and drawbacks to this approach. One advantage of the current three-sentence window approach is that a small dataset (like EDT) can still represent a larger number of writers (hence individual writing styles and stance-taking strategies) compared to using the whole document as a unit of analysis. The coverage of patterns of stance-taking strategies was deemed as important as the annotation of the entire

documents, to allow generalization of the machine learning system to different writing styles. A potential drawback of this approach is the reduction of contextual information during annotation; however, using the minimal contexts mitigates this potential issue. This point is taken up in the limitation section, where we offer recommendations and our plans for further research.

### 3.4 Core Engagement Categories

There are eight core engagement categories annotated for EDT. The category definitions and descriptions below were adapted from previous studies (Martin & White, 2005; Wu, 2007; Y. Xu, 2020). The examples are only for illustrative purposes. Note that the ***bold-italics*** in the examples show the spans to be annotated and categorized.

**Monogloss** concerns a statement that does not acknowledge any recognition of potential alternative viewpoints. Such an utterance ignores the dialogic potential in an utterance typically through bare assertions (e.g., *The language you speak **determines** your thoughts*).

**Disclaim-Deny** is an utterance that invokes an alternative position but rejects it directly (e.g., *The language you speak **does not** determine your thoughts*).

**Disclaim-Counter** is an utterance that expresses the idea so as to replace an alternative and thus counter the position which would have been expected (e.g., ***Despite the lack of evidence**, the language you speak determines your thoughts*).

**Proclaim-Concur** concerns an utterance where the writers expect/ assume that their position is easily agreed upon by the putative readers (e.g., ***As we all know**, the language you speak determines your thoughts*).

**Proclaim: Pronounce** is an utterance that shows a strong level of writer's commitment accompanied by explicit emphasis and interpolation, thereby closing down the dialogic space (e.g., *I **contend** that the language you speak determines your thoughts*).

**Proclaim: Endorse** includes utterances that use external sources as warrantable, undeniable, and/or reliable. It shows the writer's alignment with the attributed proposition (e.g., *The study by Wilson **showed** that the language you speak determines your thoughts*).

**Entertain** concerns an utterance that presents the author's position as only one possibility



amongst others, thereby opening up dialogic space (e.g., *The language you speak might influence your thoughts*).

**Attribute** concerns an utterance where the writer delegates the responsibility of a proposition to a third person (i.e., an external source), thereby opening up the dialogic space (e.g., *It is often believed that the language you speak determines your thoughts*).

It is important to reiterate that engagement is a discourse semantic category. This means that while there are some prototypical lexico-grammatical items for each category, the exact function needs to be determined with their co-text in mind, and it is challenging to create an exhaustive list of ‘expressions’ (see Hunston, 2004).

### 3.5 Supplementary discourse categories

Four supplementary discourse labels were added considering the previous discourse-analytic studies in academic domains (e.g., Hyland, 2005; Nesi, 2021). For other tags annotated, see the annotation guideline.

**Citations** is defined as mentions to an external source(s) in the text in form of in-text or narrative citation (e.g., *Smith (2000)*; *(Smith, 2000)*).

**Sources:** Mentions to an external source(s) in the text in form of nominal expressions (e.g., *A recent paper reports ...*).

**Endophoric markers** include a part of the text that refers to information in other parts of its own text (e.g., *X is discussed in Section 9*).

**Justifying** includes locutions that signal persuasion through justification or substantiation (e.g., *The current discussion is important because it highlights the key factors of climate change*).

## 4 Annotation Procedure

The annotation team consisted of two primary annotators (undergraduate students; linguistics majors) and the principal investigator (PI) (the first author, who was a Ph.D. candidate in a functional linguistics program and holds a master’s degree in Second Language Acquisition and English Language Teaching).

The annotation project comprises the following four steps—annotator training (Section 4.1), iterative consensus building (Section 4.2), independent annotation (Section 4.3), and double-checking and quality assurance (Section 4.4). The annotation comprised two tasks—detecting spans and assigning one functional label for each span.

### 4.1 Annotator Training—orientation and guided practice

Annotator training consisted of an orientation phase followed by guided practice. During the orientation phase, the two annotators were introduced to the basic concepts of SFL and the engagement system (Martin & White, 2005), which were summarized by the PI in the annotation guideline. This included the distinction between *monogloss* and *heterogloss*, the distinction between *contraction* and *expansion*, and distinct strategies (see Sections 2.2). Preliminary topics on the lexico-grammatical analysis were also reviewed as needed (Biber et al., 1999), including the notion of constituency, finite and non-finite clauses, subordinate or embedded clauses, and T-units (Hunt, 1965).

In the guided practice phase, the two annotators went through multiple-stage practice with iterative feedback from the PI. First, they were introduced to the annotation tool, WebAnno version 3.2 (Eckart de Castilho et al., 2016; Yimam et al., 2013). WebAnno was used as the graphical user interface that assists the manual span annotation of engagement resources throughout the annotation project. Second, a sample of 500 sentences was distributed to the annotators. They annotated this training sample independently, which was later checked by the IP for the mastery of the discourse annotation framework. For each annotator, the IP identified the patterns of errors in the training, provided tailored feedback independently, and clarified any concepts in the guideline. This step took the annotation team about 10 weeks (50–100 hours of working time for each annotator).

### 4.2 Iterative consensus building

In adapting the discourse-analytic framework of engagement (Martin & White, 2005), care was taken to update the annotation guidelines to make the descriptions rich and context-specific, as recommended by Fuoli (2018). To this end, the annotation team used the first 200 annotation files for active consensus building. Regular meetings were held to discuss the issues during the annotation of these files after each annotator blindly tagged the data. The resolution strategies were then documented in the annotation guideline. The initial annotation by each annotator was used for the inter-annotator agreement reported in this study.

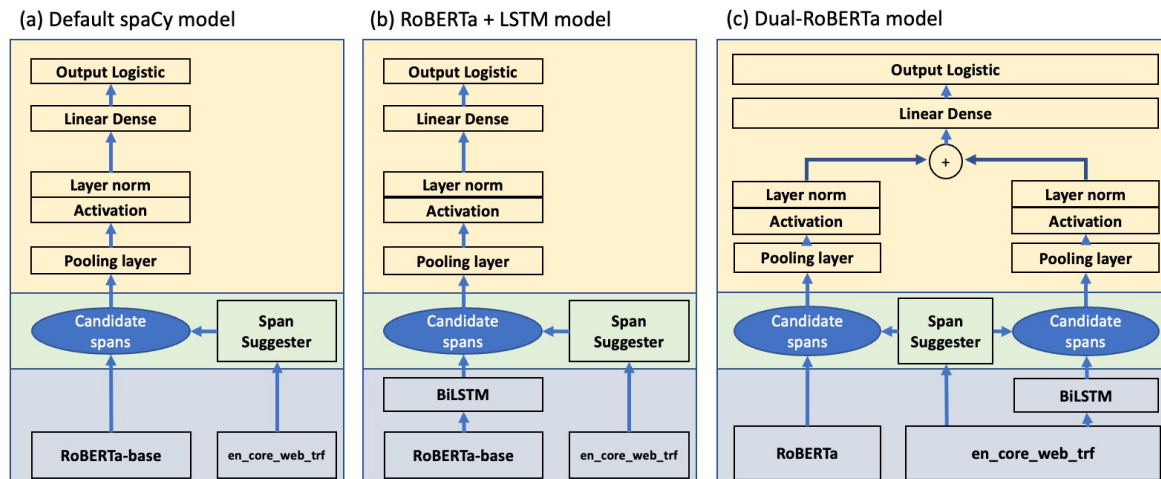


Figure 2: Three architectural variants of the proposed span identification system using the spaCy SpanCat component as the baseline (a). See Sections 5 for details.

### 4.3 Independent annotation phase

Subsequently, the two annotators were assigned to different parts of the corpus. At this point, they were encouraged to document any uncertainties in their annotations and questions in a shared spreadsheet. The annotators were allowed to ask the PI questions about ongoing issues in their annotation, which were mostly addressed in written feedback.

### 4.4 Double-check and quality assurance

Once the data is annotated by either one or two annotators, all the annotation files (both from Sections 4.2 and 4.3) were reviewed by the PI and corrected for any errors and clear deviance from the annotation guideline. After the review of each annotation file, the PI also conducted queries over the entire corpus for any inconsistencies in the spans and the categories. For example, the tag spans for *there is no X* construction (typically DENY) were inconsistently tagged ( $[no X]_{DENY}$  versus  $[there is no X]_{DENY}$ ). These inconsistencies were fixed (*there is*  $[no X]_{DENY}$  was used), and any ambiguities in the annotation guidelines were fixed for future iterations of the project.

## 5 Model Architectures

The identification and classification of engagement strategies were formulated as a span identification task (e.g., Gu et al., 2022; Lee et al., 2017). Our proposed architectures most closely resemble the span enumeration approach in Gu (2022), where candidate spans are generated greedily (using  $n$ -

grams and dependency subtrees). Figure 2 shows three variants of our neural architecture. We started from the baseline spaCy span categorizer model (Honnibal et al., 2020). We then gradually built the model complexity, guided by previous work in span identification (e.g., Gu et al., 2022; Lee et al., 2017; Papay et al., 2020; Zhu et al., 2021) and our intuitions as linguists. The basic span categorizer pipeline consists of Token embedder, Span Suggester, and Span Categorizer.

### 5.1 Baseline—spaCy Span Categorizer

The first group of ML models uses a single transformer layer as Token Embedder, which is then sent to a pooling layer and logistic regression (see diagram [a] in the Figure 2). This is the default span categorizer implementation provided by spaCy (Honnibal et al., 2020; Schmuhl et al., 2022). In our implementations, we used the off-the-shelf spaCy `en_core_web_trf` model to predict dependency representations of the input text, which were used to suggest candidate spans along with  $n$ -grams. For each candidate span, span representation is created by taking the RoBERTa-base embeddings and applying several pooling operations. The pooled span representation is sent to the non-linear activation function and subsequently to the logistic layer for prediction. In this architecture, the RoBERTa embeddings were fine-tuned to learn task-specific weights while the weights from the `en_core_web_trf` model were fixed.

## 5.2 RoBERTa + Bi-LSTM model

Although Transformer models can provide contextually aware token representations (Clark et al., 2019), it was hypothesized that additional sequential information would be beneficial for classifying engagement strategies that are interpreted by discourse analysts with co-textual information in mind, such as CONTRIBUTION. To allow the model to learn this additional contextual information, we added a single-layer Bidirectional Long-Short Term Memory (Bi-LSTM; Hochreiter & Schmidhuber, 1997; Schuster & Paliwal, 1997) architecture on top of the RoBERTa token-level embeddings, before they are sent to the span pooling layer. Such architecture has often been implemented in previous span identification architectures (Gu et al., 2022; Lee et al., 2017; Papay et al., 2020; Zhu et al., 2021). For the purpose of the current study, we used one-layer Bi-LSTM with 200 hidden dimensions following the previous study (Gu et al., 2022).

## 5.3 Dual-RoBERTa model

The third architecture used two sets of transformer embeddings side-by-side, concatenated before the final output layer for prediction (see Architecture (c) in Figure 2). This model architecture was inspired by recent ensemble approaches to span identification pipelines (e.g., Rao, 2022). The intuition behind the dual-Transformer architecture was that the two Transformer models would offer complementary information to categorize the span labels, particularly because the second Transformer layer from the spaCy `en_core_web_trf` model was already fine-tuned for multitask learning objectives (e.g., POS tagging, Dependency parsing, Named Entity Recognition) on the Ontonote 5.0 corpus (Weischedel et al., 2013). Note that the RoBERTa weights from the `en_core_web_trf` was fixed in order to avoid forgetting of the important information for the dependency parsing.

## 5.4 Domain adaptation of RoBERTa

Since the version of EDT used for training was still relatively small, adaptive pre-trainings were conducted on the RoBERTa-base model (Liu et al., 2019) using the checkpoint available through Hugging Face library (Wolf et al., 2020) in hope to counteract potential mismatches between the RoBERTa embedding and the characteristics of in-domain texts (Han & Eisenstein, 2019; Ramponi & Plank, 2020). To this end, four domain adapted

RoBERTa-base models were created. The five versions of RoBERTa (including the original) were set as hyperparameter in the following experiment (see Appendix B).

## 6 Methods

We implemented the three architectures through spaCy version 3.4 (Honnibal et al., 2020). All models were trained on a quad Nvidia Tesla K80 GPU with 12GB RAM. All models were optimized with Adam Optimizer.

### 6.1 Data preparation

Table 1 summarizes the number of tags by category in the dataset used for this experiment. Two pairs of tags (Concur and Pronounce; Endorse and Attribute) were collapsed as PROCLAIM and CONTRIBUTION, respectively, to obtain enough number of instances in the dev and test sets. According to the engagement system (Martin & White, 2005), Concur and Pronounce are subtypes of PROCLAIM strategy along with ENDORSE, while ENDORSE was categorized under CONTRIBUTION with Attribute in this study due to its primary function of such (Sections 2.2 and 3.4). We then created five sets of 80/10/10 splits for 5-fold cross-validations (CV). The tag counts in the 5-fold datasets can be found in Appendix A. Due to the imbalances in labels, we oversampled minority cases in each data split (after splitting them into training sets to avoid data leaks). The oversampling approach (e.g., Wang & Wang, 2022) was used because there is no existing model to create synthetic examples for this new type of NLP task.

Category	Tag counts
CONTRIBUTION	1247
COUNTER	1046
DENY	887
ENTERTAIN	2837
MONOGLOSS	2742
PROCLAIM	445
CITATION	618
ENDOPHORIC	213
JUSTIFYING	966
SOURCES	855

Table 1: The number of tags by category in the entire EDT. CONTRIBUTION subsumes CONTRIBUTION and ENDORSE; PROCLAIM subsumes CONCUR and PRONOUNCE in the original tags.

Category	Human annotation baselines		End-to-end models trained on adjudicated data					
	Read & Carroll (2012)	Our annotator agreement	spaCy default		RoBERTa+LSTM		Dual-RoBERTa	
			<i>M</i>	<i>Min</i>	<i>M</i>	<i>Min</i>	<i>M</i>	<i>Min</i>
ATTRIBUTION	.379	.5943	.6969	.6553	<b>.7127</b>	.6761	.6911	.6149
COUNTER	.603	.8511	.8521	.7394	.8636	.7781	<b>.8774</b>	.8567
DENY	.451	.8621	.8570	.8257	.8800	.8579	<b>.8815</b>	.8522
ENTERTAIN	.459	.8278	.8413	.7917	<b>.8360</b>	.7755	.8340	.7903
MONOGLOSS	n/a	.8092	<b>.8017</b>	.7476	.7864	.7568	.7890	.7314
PROCLAIM	.336	.4038	.6685	.6127	.6906	.6203	<b>.7027</b>	.6197
CITATION	n/a	.9497	.9047	.8875	.9185	.8953	<b>.9193</b>	.9015
ENDOPHORIC	n/a	.6071	.7236	.6000	.7254	.6316	<b>.7418</b>	.6919
JUSTIFYING	n/a	.8203	.8131	.7766	<b>.8167</b>	.7404	.8081	.7608
SOURCES	n/a	.5663	.6961	.6585	<b>.6985</b>	.6318	.6844	.5887
Accuracy		.7146	.7015	.6885	<b>.7095</b>	.6960	.7054	.6922
macro avg F1		.6629	.7141	.6942	.7208	.7105	<b>.7209</b>	.7108
weighted avg F1		.7208	.7183	.7094	<b>.7283</b>	.7105	.7196	.6903
Cohen's Kappa		.6686	.6647	.6509	<b>.6738</b>	.6596	.6694	.6549
MMC		.6691	.6663	.6534	<b>.6755</b>	.6611	.6710	.6554

Table 2: F1 scores based on 5-Fold CV. Our intercoder agreement is presented side by side with the result reported in Read and Carroll (2012), who annotated the entire Appraisal framework. Due to the adaptations of the original Martin and White (2005) in our study (see Section 3.4) some of the tags lacks direct comparisons. Three neural architectures are compared using the mean and minimum F1 scores based on the 5-Fold CV. MCC = Matthews Correlation Coefficient. Averaged F1 scores were calculated including empty tags.

## 6.2 Hyperparameter tuning and 5-fold cross-validation

We randomly searched the optimal combination of hyperparameters for each of the three architectures and tested the stabilities of the top three settings from each architecture (see Appendix B for hyperparameters). A total of 205 models were trained across the three architectures. Subsequently, eight top-performing hyperparameter settings were chosen, and we then conducted 5-fold cross-validation for each. We report the result of the best 5-fold CV result for each architecture.

## 6.3 Evaluation metrics

Considering the imbalanced data, the models were evaluated using Matthews Correlation Coefficient and Cohen's Kappa on the end-to-end span categorization results. Because our span suggester used span enumeration approach (Gu et al., 2022) and was constant across the models, they were not compared. Note that preliminary experiments showed that the current span suggester (See Appendix B for hyperparameter settings) achieved recall of 97–99% on Development and Test sets.

## 7 Results

Table 2 reports on the inter-annotator reliability and the results of the 5-fold CV.

### 7.1 Inter-annotator agreement

A subset of blind annotation (35,640 tokens; 1,373 sentences; 3,732 unique spans) was used to compute the inter-annotator agreement between the two annotators. The results indicated that the agreement was moderate (Cohen's Kappa = .6686; Matthews Correlation Coefficient = .6691). Comparing the by-tag F1 scores against those by Read and Carroll (2012), our annotator agreement was substantially higher. However, the results also indicate there were some areas of struggle by human annotators (e.g., ATTRIBUTION, PROCLAIM).

### 7.2 Result of the end-to-end models

Overall, the end-to-end models, which were trained on a fully reviewed/adjudicated dataset, tended to outperform the benchmarks of inter-annotator agreement. The gains were substantial in several categories that were challenging for our annotators, including ATTRIBUTION, PROCLAIM, and SOURCES.

### 7.3 Comparison among three architectures

The result of the 5-fold CV (Table 2) indicated that the RoBERTa + LSTM architecture performed best among the three architectures (Cohen's Kappa = .6738; Matthews Correlation Coefficient = .6755). This was followed by the Dual-RoBERTa Model (Cohen's Kappa = .6694; Matthews Correlation Coefficient = .6710). It appears that RoBERTa + LSTM model and Dual-RoBERTa model may complement their strengths and weaknesses.

## 8 Discussion

The results of the 5-fold CVs indicated that the proposed architectures performed as well as (or even outperformed) the inter-annotator agreement baseline set for the study. The results also suggested that our RoBERTa + LSTM and Dual-RoBERTa models tended to perform better than the spaCy default spancat model (Honnibal et al., 2020; Schmuhl et al., 2022).

It is noteworthy that the additional Bi-LSTM layer appeared to enhance the stability of the model. Although the use of a Bi-LSTM layer on top of a Transformer encoder is not uncommon in span identification tasks, its reported benefits have been mixed (Gu et al., 2022; Papay et al., 2020; Zhu et al., 2021). The gain in this study can be explained in two ways—additional sequential information and dimensional reduction. In a simple explanation, the architecture benefited from the additional sequence information provided by Bi-LSTM. At least one previous study (Gu et al., 2022) reported similar gains in additional LSTM layer, particularly when the span suggestion components were similar to the current greedy approach. Thus, it could be that the additional LSTM helped to refine the embedding for this particular span enumeration architecture (Gu et al., 2022; Lee et al., 2017). In addition to this explanation, it is also possible that LSTM worked as a dimension reducer (while maintaining direct sequential information). Future research may clarify the potential reasons for this stability in the span identification task (which is out of the scope of the current study).

Apart from the machine learning experiment, our inter-annotator agreement showed that the span annotation of engagement resource may be a challenging task, particularly for undergraduate annotators (linguistics majors) who were trained over 10 weeks. However, our annotator agreement

substantially improved upon the previously published benchmark by Read and Carroll (2012). The moderate reliability in this study may provide further evidence to Fuoli's (2018) claim regarding the lack of explicit guidelines and methodological discussions pertaining to the identification of engagement resources in discourse samples. Thus, it is hoped that the present annotation guideline may serve as a resource to guide future methodological improvement in discourse annotation of engagement resource analysis (see Fuoli, 2018; Read & Carroll, 2012).

## 9 Conclusion

In this paper, we reported a new approach to identifying stance-taking expressions in English texts in academic domains. Specifically, we introduced a new human-annotated corpus of academic English that draws on a discourse-analytic framework of the engagement system from the Appraisal framework (Martin & White, 2005). We also reported an end-to-end system that can conduct automated span identification of stance-taking strategies based on the engagement framework. The experimental result indicates that the system can outperform inter-annotator reliability estimates by a 5–6% gain in the macro-averaged F1 score. The finding, although preliminary, opens a new avenue for feature engineering for the next-generation AWE systems (Burstein et al., 2016), expanding the constructs measured by the AWE engines. A follow-up study by the first author shows that the engagement features can explain the writing scores above and beyond the existing linguistic features at the levels of lexis, grammar, and cohesion (Eguchi, 2023). While end-to-end score prediction models may be used to obtain accurate score predictions, the features introduced in this paper may be used in conjunction with such end-to-end scoring engines to maintain the explainability and interpretability of the scores. The visualization of stance-taking features (see Figure 1 and demo) can also be presented to learners to highlight the patterns of stance-taking in both model and student essays.

### 9.1 Limitations and Future Directions

For future work, we plan to update the annotated corpus and the scope of annotation to paragraphs and/or whole documents. In this study, we opted for the minimal context approach for practical reasons, such as budget, time constraints, and copy rights of

the source corpora. The minimal context approach allowed the annotation sample to represent as many writers as possible for better generalization with relatively small sample sizes. However, future research should use longer units of analysis to enhance the quality of manual annotation. Despite this limitation, the results of the current study indicated that the current approach is a promising direction for further research on automated analyses of rhetorical features.

## Acknowledgments

We thank three anonymous reviewers for their insightful comments, which improved the quality of the current manuscript. We express our sincere gratitude to two undergraduate annotators, Aaron Miller and Ryan Walker, for their meticulous work on the discourse annotation. This work was supported by the following grants/awards: the Duolingo English Test's Doctoral Dissertation Award 2022, the International Research Foundation for English Language Education (TIRF) Doctoral Dissertation Grant 2022, the National Federation of Modern Language Teachers Association and the Modern Language Journal (NFMLTA-MLJ) Dissertation Writing Support Grant 2022, the Graduate Student Research Award at the Linguistics Department, University of Oregon, and Dr. Kristopher Kyle's institutional research funds.

## References

- Alsop, S., & Nesi, H. (2009). Issues in the development of the British Academic Written English (BAWE) corpus. *Corpora*, 4(1), 71–83. <https://doi.org/10.3366/E1749503209000227>
- Attali, Y. (2007). Construct validity of e-rater® in scoring TOEFL® essays. *ETS Research Report Series*, 2007(1), i–22. <https://doi.org/10.1002/j.2333-8504.2007.tb02063.x>
- Bachman, L. F., & Palmer, A. S. (1996). *Language testing in practice: Designing and developing useful language tests*. Oxford University Press.
- Bachman, L. F., & Palmer, A. S. (2010). *Language assessment in practice: Developing language assessments and justifying their use in the real world*. Oxford University Press.
- Bax, S., Nakatsuhara, F., & Waller, D. (2019). Researching L2 writers' use of metadiscourse markers at intermediate and advanced levels. *System*, 83, 79–95. <https://doi.org/10.1016/j.system.2019.02.010>
- Biber, D. (2006). Stance in spoken and written university registers. *Journal of English for Academic Purposes*, 5(2), 97–116. <https://doi.org/10.1016/j.jeap.2006.05.001>
- Biber, D., & Finegan, E. (1988). Adverbial stance types in English. *Discourse Processes*, 11(1), 1–34. <https://doi.org/10.1080/01638538809544689>
- Biber, D., Johansso, S., Leech, G., Conrad, S., & Finegan, E. (Eds.). (1999). *Longman grammar of spoken and written English* (10. impression). Longman.
- Blanchard, D., Tetreault, J., Higgins, D., Cahill, A., & Chodorow, M. (2013). TOEFL11: A Corpus of Non-Native English. *ETS Research Report Series*, 2013(2), i–15. <https://doi.org/10.1002/j.2333-8504.2013.tb02331.x>
- Burstein, J., Elliot, N., Klebanov, B. B., Madnani, N., Napolitano, D., Schwartz, M., Houghton, P., & Molloy, H. (2018). Writing MentorTM: Writing Progress Using Self-Regulated Writing Support. *The Journal of Writing Analytics*, 2(1), 285–313. <https://doi.org/10.37514/JWA-J.2018.2.1.12>
- Burstein, J., Elliot, N., & Molloy, H. (2016). Informing Automated Writing Evaluation Using the Lens of Genre: Two Studies. *CALICO Journal*, 33(1), 117–141. <https://doi.org/10.1558/cj.v33i1.26374>
- Carr, N. T. (2013). Computer-automated scoring of written responses. In A. J. Kunnan (Ed.), *The Companion to Language Assessment* (pp. 1063–1078). John Wiley & Sons, Inc. <https://doi.org/10.1002/9781118411360.wbcla124>
- Chang, P., & Schleppegrell, M. (2011). Taking an effective authorial stance in academic writing: Making the linguistic resources explicit for L2 writers in the social sciences. *Journal of English for Academic Purposes*, 10(3), 140–151. <https://doi.org/10.1016/j.jeap.2011.05.005>
- Clark, K., Khandelwal, U., Levy, O., & Manning, C. D. (2019). What Does BERT Look At? An Analysis of BERT's Attention. *ArXiv:1906.04341 [Cs]*. <http://arxiv.org/abs/1906.04341>
- Cotos, E. (2014). *Genre-Based Automated Writing Evaluation for L2 Research Writing*. Palgrave Macmillan UK. <https://doi.org/10.1057/9781137333377>
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv:1810.04805 [Cs]*. <http://arxiv.org/abs/1810.04805>
- Eguchi, M. (2023). *Automatic Analysis of Epistemic Stance-Taking in Academic English Writing: A Systemic Functional Approach* [Doctoral dissertation]. University of Oregon.

- Fiacco, J., Jiang, S., Adamson, D., & Rosé, C. (2022). Toward Automatic Discourse Parsing of Student Writing Motivated by Neural Interpretation. *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, 204–215. <https://doi.org/10.18653/v1/2022.bea-1.25>
- Fu, J., Huang, X., & Liu, P. (2021). SpanNER: Named Entity Re-/Recognition as Span Prediction. *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)*, 7183–7195. <https://doi.org/10.18653/v1/2021.acl-long.558>
- Fuoli, M. (2018). A stepwise method for annotating appraisal. *Functions of Language*, 25(2), 229–258. <https://doi.org/10.1075/fof.15016.fuo>
- Gu, W., Zheng, B., Chen, Y., Chen, T., & Van Durme, B. (2022). *An Empirical Study on Finding Spans* (arXiv:2210.06824). [arXiv: http://arxiv.org/abs/2210.06824](http://arxiv.org/abs/2210.06824)
- Han, X., & Eisenstein, J. (2019). Unsupervised Domain Adaptation of Contextualized Embeddings for Sequence Labeling. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 4238–4248. <https://doi.org/10.18653/v1/D19-1433>
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Honnibal, M., Ines, M., Van Landeghem, S., & Boyd, A. (2020). *spaCy: Industrial-strength Natural Language Processing in Python* (3.3). <https://spacy.io>
- Huawei, S., & Aryadoust, V. (2023). A systematic review of automated writing evaluation systems. *Education and Information Technologies*, 28(1), 771–795. <https://doi.org/10.1007/s10639-022-11200-7>
- Hunston, S. (2004). Counting the uncountable: Problems of identifying evaluation in a text and in a corpus. In A. S. Partington (Ed.), *Corpora and Discourse*.
- Hunston, S., & Thompson, G. (2000). *Evaluation in Text: Authorial Stance and the Construction of Discourse*. Oxford University Press, UK.
- Hunt, K. W. (1965). Grammatical structures written at three grade levels. *NCTE Research Report No. 3*. <http://eric.ed.gov/?id=ED113735>
- Hyland, K. (2005). *Metadiscourse: Exploring interaction in writing*. Continuum.
- Ishikawa, S. (2013). *The ICNALE and Sophisticated Contrastive Interlanguage Analysis of Asian Learners of English*. 神戸大学国際コミュニケーションシヨウセンセンター. <https://doi.org/10.24546/81006678>
- Kyle, K. (2020). The relationship between features of source text use and integrated writing quality. *Assessing Writing*, 45, 100467. <https://doi.org/10.1016/j.asw.2020.100467>
- Lam, S. L., & Crosthwaite, P. (2018). APPRAISAL resources in L1 and L2 argumentative essays: A contrastive learner corpus-informed study of evaluative stance. *Journal of Corpora and Discourse Studies*, 1(1), 8. <https://doi.org/10.18573/jcads.1>
- Lancaster, Z. (2014). Exploring valued patterns of stance in upper-level student writing in the disciplines. *Written Communication*, 31(1), 27–57. <https://doi.org/10.1177/0741088313515170>
- Lee, K., He, L., Lewis, M., & Zettlemoyer, L. (2017). End-to-end Neural Coreference Resolution. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 188–197. <https://doi.org/10.18653/v1/D17-1018>
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Michael Lewis, Michael Lewis, Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv: Computation and Language*.
- Lu, X. (2021). Directions for future automated analyses of L2 written texts. In R. M. Manchón & C. Polio, *The Routledge handbook of second language acquisition and writing* (1st ed., pp. 370–382). Routledge. <https://doi.org/10.4324/9780429199691-36>
- Martin, J. R., & White, P. R. R. (2005). *The language of evaluation: Appraisal in English*. Palgrave Macmillan.
- Nesi, H. (2021). Sources for courses: Metadiscourse and the role of citation in student writing. *Lingua*, 253, 103040. <https://doi.org/10.1016/j.lingua.2021.103040>
- Papay, S., Klinger, R., & Padó, S. (2020). Dissecting Span Identification Tasks with Performance Prediction. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 4881–4895. <https://doi.org/10.18653/v1/2020.emnlp-main.396>
- Pareti, S. (2016). PARC 3.0: A Corpus of Attribution Relations. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 3914–3920. <https://aclanthology.org/L16-1619>

- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1), 140:5485-140:5551.
- Ramponi, A., & Plank, B. (2020). *Neural Unsupervised Domain Adaptation in NLP---A Survey* (arXiv:2006.00632). arXiv. <https://doi.org/10.48550/arXiv.2006.00632>
- Rao, A. R. (2022). ASRtrans at SemEval-2022 Task 4: Ensemble of Tuned Transformer-based Models for PCL Detection. *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, 344–351. <https://doi.org/10.18653/v1/2022.semeval-1.44>
- Read, J., & Carroll, J. (2012). Annotating expressions of Appraisal in English. *Language Resources and Evaluation*, 46(3), 421–447. <https://doi.org/10.1007/s10579-010-9135-7>
- Römer, U., & O'Donnell, M. B. (2011). From student hard drive to web corpus (part 1): The design, compilation and genre classification of the Michigan Corpus of Upper-level Student Papers (MICUSP). *Corpora*, 6(2), 159–177.
- Schiller, B., Daxenberger, J., & Gurevych, I. (2021). Stance Detection Benchmark: How Robust is Your Stance Detection? *KI - Künstliche Intelligenz*, 35(3), 329–341. <https://doi.org/10.1007/s13218-021-00714-w>
- Schmuhl, E., Miranda, L., Kádár, Á., Van Landeghem, S., & Boyd, A. (2022, June 14). *Spancat: A new approach for span labeling · Explosion*. Explosion. <https://explosion.ai/blog/spancat>
- Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11), 2673–2681. <https://doi.org/10.1109/78.650093>
- Shermis, M. D., & Burstein, J. (Eds.). (2013). *Handbook on automated essay evaluation: Current applications and new directions*. Routledge.
- Sparks, J. R., Song, Y., Brantley, W., & Liu, O. L. (2014). Assessing Written Communication in Higher Education: Review and Recommendations for Next-Generation Assessment. *ETS Research Report Series*, 2014(2), 1–52. <https://doi.org/10.1002/ets2.12035>
- Wang, X., & Wang, Y. (2022). Sentence-Level Resampling for Named Entity Recognition. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2151–2165. <https://doi.org/10.18653/v1/2022.naacl-main.156>
- Weischedel, R., Palmer, M., Marcus, M., Hovy, E., Pradhan, S., Ramshaw, L., Xue, N., Taylor, A., Kaufman, J., & Franchini, M. (2013). *Ontonotes release 5.0*. Linguistic Data Consortium.
- White, P. R. R. (2003). Beyond modality and hedging: A dialogic view of the language of intersubjective stance. *Text - Interdisciplinary Journal for the Study of Discourse*, 23(2). <https://doi.org/10.1515/text.2003.011>
- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., ... Rush, A. M. (2020). *HuggingFace's Transformers: State-of-the-art Natural Language Processing* (arXiv:1910.03771). arXiv. <http://arxiv.org/abs/1910.03771>
- Wu, S. M. (2007). The use of engagement resources in high- and low-rated undergraduate geography essays. *Journal of English for Academic Purposes*, 6(3), 254–271. <https://doi.org/10.1016/j.jeap.2007.09.006>
- Xie, J. (2020). A review of research on authorial evaluation in English academic writing: A methodological perspective. *Journal of English for Academic Purposes*, 47, 100892. <https://doi.org/10.1016/j.jeap.2020.100892>
- Xu, X., & Nesi, H. (2017). An analysis of the evaluation contexts in academic discourse. *Functional Linguistics*, 4(1). <https://doi.org/10.1186/s40554-016-0037-x>
- Xu, Y. (2020). *SECOND LANGUAGE WRITING COMPLEXITY IN ACADEMIC LEGAL DISCOURSE: DEVELOPMENT AND ASSESSMENT UNDER A CURRICULAR LENS* [PhD Dissertation]. Georgetown University.
- Yannakoudakis, H., Briscoe, T., & Medlock, B. (2011). A New Dataset and Method for Automatically Grading ESOL Texts. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 180–189. <https://aclanthology.org/P11-1019>
- Yoon, H.-J. (2017). Linguistic complexity in L2 writing revisited: Issues of topic, proficiency, and construct multidimensionality. *System*, 66, 130–141. <https://doi.org/10.1016/j.system.2017.03.007>
- Zhu, Q., Lin, Z., Zhang, Y., Sun, J., Li, X., Lin, Q., Dang, Y., & Xu, R. (2021). HITSZ-HLT at SemEval-2021 Task 5: Ensemble Sequence Labeling and Span Boundary Detection for Toxic Span Detection. *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, 521–526. <https://doi.org/10.18653/v1/2021.semeval-1.63>



### A Tag counts in each training fold (Before oversampling)

	Fold 1			Fold 2			Fold 3			Fold 4			Fold 5		
	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test
ATTRIBUTION:	1028	118	101	995	136	116	993	115	139	987	138	122	985	129	133
COUNTER:	879	88	79	818	112	116	839	105	102	848	103	95	800	127	119
DENY:	705	83	99	712	86	89	712	84	91	720	94	73	699	88	100
ENTERTAIN:	2306	254	277	2262	314	261	2246	280	311	2310	248	279	2224	334	279
MONOGLOSS:	2223	256	263	2179	264	299	2184	289	269	2157	296	289	2225	273	244
PROCLAIM:	343	47	55	353	54	38	358	33	54	375	33	37	351	41	53
ENDOPHORIC:	168	29	16	175	18	20	176	12	25	168	30	15	165	21	27
JUSTIFYING:	788	84	94	783	93	90	750	118	98	795	79	92	748	108	110
CITATION:	517	56	45	497	56	65	504	48	66	479	71	68	475	60	83
SOURCES:	700	76	79	686	85	84	682	79	94	660	111	84	692	87	76
sum	9657	1091	1108	9460	1218	1178	9444	1163	1249	9499	1203	1154	9364	1268	1224

## B Hyperparameters for random search

Category	Hyperparameter	Possible values (Parameter range or choice)	Selection
Entire model	Model Architecture	Single-Transformer; Single-Transformer+ LSTM; Dual-Transformer + single-LSTM	discrete
Token Embedder	Pre-Trained language model	<a href="#">roberta-base</a> ; <a href="#">egumasa/roberta-base-academic3</a> ; <a href="#">egumasa/roberta-base-university-writing2</a> ; <a href="#">egumasa/roberta-base-research-papers</a>	discrete
Span Categorizer	FFN (Activation function)	Maxout (default selection by spaCy); Mish; Mish with two separate FFNs	discrete
Span Categorizer	FFN (hidden unit sizes)	[128, 256, 384]	discrete
Span Categorizer	FFN (dropout rates)	[0, 0.2, 0.3, 0.4]	discrete
Span Categorizer	FFN (layer depths)	[1, 2]	discrete
Training	Maximum learning rate (alpha)	6e-5 – 2e-5	uniform distribution
Training	System seed during training	[0, 808, 1993, 1234, 2023]	discrete
Training	Gradient accumulation steps	[4, 8]	discrete
Span Suggester	Max n-gram lengths	12 words	fixed
Training	Optimizer	Adam with weight decay	fixed
Training	Learning rate schedule	linear decay with warm-up steps	fixed
Training	Warm-up steps	1,000	fixed
Training	Maximum training step	20,000	fixed
Training	Steps before early stop	3,000	fixed
Training	mini-batch size	defined by number of words	fixed
Training	minimal start batch size	[300, 500, 900]	discrete
Training	Maximum batch size	1,000 words	fixed

# ACTA: Short-Answer Grading in High-Stakes Medical Exams

King Yiu Suen<sup>1</sup>, Victoria Yaneva<sup>1</sup> Le An Ha<sup>2</sup> Janet Mee<sup>1</sup>,

Yiyun Zhou<sup>1</sup>, Polina Harik<sup>1</sup>

<sup>1</sup>National Board of Medical Examiners, Philadelphia, USA

{ksuen, vyaneva, jmee, yyzhou, pharik}@nbme.org

<sup>2</sup>University of Wolverhampton, UK

ha.l.a@wlv.ac.uk

## Abstract

This paper presents the ACTA system, which performs automated short-answer grading in the domain of high-stakes medical exams. The system builds upon previous work on neural similarity-based grading approaches by applying these to the medical domain and utilizing contrastive learning as a means to optimize the similarity metric. ACTA is evaluated against three strong baselines and is developed in alignment with operational needs, where low-confidence responses are flagged for human review. Learning curves are explored to understand the effects of training data on performance. The results demonstrate that ACTA leads to substantially lower number of responses being flagged for human review, while maintaining high classification accuracy.

## 1 Introduction

Automated Short Answer Grading (ASAG) has been a longstanding educational application of NLP. The task of classifying the free-text responses to short-answer questions (SAQs) as *correct* or *incorrect* is made challenging by the fact that the same concept may be expressed in a myriad of different ways. The problem has received considerable attention, with several competitions organized on the topic such as a SemEval shared task by Dzikovska et al. (2013) or the ASAP 2 Kaggle competition<sup>1</sup>.

Most broadly, the ASAG literature defines two scoring approaches: an *instance-based approach*, where a system is trained on a portion of the data and outputs a predicted score for a given new response, and a *similarity-based approach*, where each new response assumes the label of an annotated response it is matched to using some similarity metric (Bexte et al., 2022). In early work, pre-neural similarity-based approaches were shown to lag behind the less interpretable instance-based approaches (Sakaguchi et al., 2015). Since then,

neural similarity-based approaches have shown increasing promise by learning response (or question-response) embeddings and matching the pairs using cosine similarity (e.g. Schneider et al. (2022)). Bexte et al. (2022) proposed that the similarity-based approach can be further improved if the similarity metric is appropriately optimized. In their work, a pretrained Sentence-BERT model (Reimers and Gurevych, 2019) is fine-tuned on answer pairs and then a k-nearest neighbors classifier is used to match a new response based on its similarity to the labeled ones. These advances have led to a considerable improvement over the instance-based approach not only in terms of accuracy, but also in terms of interpretability and the need for less annotated data for training.

In this study, we present the ACTA system (Analysis of Clinical Text for Assessment), where we build upon the work of Bexte et al. (2022) by exploring the use of contrastive learning (Chopra et al., 2005) as a way to optimize the performance of similarity-based approaches and by applying the approach to the clinical domain. The contributions of this paper are as follows:

- Exploration of the similarity-based ASAG approach in the clinical domain, which is characterized by a number of challenging idiosyncrasies such as complex terminology, extensive use of abbreviations, misspellings, etc.
- Comparison of the results to three baselines: majority class, a similarity-based approach without finetuning, and a previous scoring system designed for the clinical domain.
- System and evaluation design constructed in alignment with operational needs, where responses that do not satisfy a given confidence threshold are flagged for human review.
- Exploration of learning curves with various training set sizes, as well as experimentation with various confidence thresholds.

<sup>1</sup><https://www.kaggle.com/c/asap-sas>

## 2 Data

We perform experiments on two datasets containing short free-text responses to clinical test items.

Set 1 consists of SHARP items (Short Answer Rationale Provision items) – an item format where examinees see a patient chart and are asked to provide a free-text response regarding the most likely diagnosis (e.g., “plantar fasciitis”, “dermatomyositis”), most appropriate next steps (e.g., “Administer corticosteroids then do arterial biopsys”), causes (e.g., “Homocysteine and MMA levels in blood”), etc.<sup>2</sup> A total of 44 items were administered in a pilot involving 177 4th-year US medical students. Each student saw each item, resulting in a total of 7,788 responses (of which 2,807 were unique).

Set 2 consists of short-answer questions, which present a vignette<sup>3</sup> describing a clinical case. Similar to Set 1, the Set 2 responses included diagnoses, causes, and treatments, among other categories of responses. These items were administered to 8,162 US medical students as part of their Internal Medicine school subject exam. There were 71 Set 2 items, where each item was seen by an average of 176 examinees (SD = 12.620), resulting in a total of 12,508 free-text responses (5,696 unique).

Responses from both sets were scored as *correct* or *incorrect* by content experts (physicians and nurse practitioners) using a scoring rubric for each item. For Set 1, two subject matter experts scored the items together as part of developing scoring guidelines for future pilots (hence agreement statistics for independent scoring cannot be reported). Another group of physicians reviewed the scores and confirmed agreement with the scoring procedure. For Set 2, four judges scored the items. Kappa coefficients (based on unique responses) for the six possible pairs of judges ranged from 0.89 to 0.92, indicating strong agreement. Scoring resulted in 5,201 correct responses (66.78%) for Set 1 and 8,086 (64.64%) for Set 2.

## 3 Method

We use contrastive representation learning (Chopra et al., 2005) to encode responses into embedding vectors such that responses with the same score have similar embeddings and responses with dif-

ferent scores have very different ones. For any given two responses, the degree to which they are matched can then be measured by the cosine similarity between their embedding vectors. Similar to Bexte et al. (2022), we use Sentence-BERT (a.k.a. SBERT) to derive the embeddings for each response, since the model introduces a modification of the pretrained BERT network that “reduces the effort for finding the most similar pair from 65 hours with BERT / RoBERTa to about 5 seconds” (Reimers and Gurevych, 2019).

First, we pair up every response with every other response for the same item. Each pair is assigned a label of 1 if both responses have the same score (both correct or both incorrect), 0 otherwise. For each pair, the two responses are passed to SBERT independently, producing two sentence embedding vectors (one for each response).

The contrastive loss encourages the model to minimize the embedding distance when responses have the same score, and maximize the distance otherwise. To do that, the cosine similarity and the cosine distance between the sentence embedding of the first response  $e_1$  and the sentence embedding of the second response  $e_2$  are defined as:

$$\text{similarity}(e_1, e_2) = \frac{e_1^T \cdot e_2}{\|e_1\| \|e_2\|}$$

$$\text{distance}(e_1, e_2) = 1 - \text{similarity}(e_1, e_2)$$

Then, the contrastive loss is defined as

$$\mathcal{L}(e_1, e_2, \text{label}) = \text{label} \cdot (\text{distance}(e_1, e_2))^2 + (1 - \text{label}) \cdot \max(0, \text{margin} - \text{distance}(e_1, e_2))^2$$

where margin is a hyperparameter, defining the lower bound distance between responses with different scores. One advantage of contrastive loss over cosine similarity loss is that it goes to 0 for negative pairs when the distance is farther than the margin. When dissimilar inputs are sufficiently distant there is no more pressure on the model to keep pushing them apart, which could allow the model to focus on improving the most erroneous cases.

During inference, the trained model is used to compute the cosine similarity between the sentence embedding of the new response and the sentence embedding of every annotation (i.e., responses of the same item in the training set). If the highest

<sup>2</sup>Other aspects of the SHARP item format that refer to subsequent steps for measuring clinical reasoning are not described here.

<sup>3</sup>See Ha et al. (2020) for a detailed description of the use of vignette-based SAQs in medicine.

	Training	Set 1 (SHARP items)						Set 2 (SAQs)			
		20	40	60	80	120	142	20%	40%	60%	80%
INCITE	F1	.986	.986	.989	.984	.988	.989	.88	.9	.88	.882
	Unmatched	488	442	397	354	334	318	987	830	748	711
ACTA No Finetuning	F1	.998	.998	1.	1.	1.	1.	.999	.999	.999	.997
	Unmatched	623	523	463	429	385	368	970	835	743	684
ACTA Finetuned	F1	.995	.993	.977	.979	.982	.982	.991	.991	.978	.972
	Unmatched	545	443	201	123	47	44	734	497	274	172

Table 1: Results for a similarity threshold of .95, where "F1" indicates classification performance for all matched items and "Unmatched" indicates the number of items that need to go through human scoring. For Set 1, the training data size is measured in number of examinees whose data was used for training (e.g., the first 20 examinees, the first 40, etc.). In Set 2, it is measured as percentage of the full dataset. Note that for ACTA No Finetuning, the term "training set" refers to the subset of data used to identify the most similar instances for a given new response.

cosine similarity is less than a given threshold, the new response is labeled as *unmatched* and flagged for human rater review. Otherwise, the new response assumes the score of the annotation that it has the highest cosine similarity with. For detailed training parameters, see Appendix A.

#### 4 Experimental setup

**Baselines:** We compare the approach proposed in ACTA to three baselines: **a majority class baseline** (always predicting a *correct* response); **ACTA No finetuning** – a similarity-based approach using SBERT, where the model was not trained to optimize the similarity metric. We use *all-MiniLM-L6-v2*<sup>4</sup>, which has been pretrained on 1B sentence pairs, as our backbone model for both SBERT-no-training and SBERT. Finally, **the INCITE system** (Sarker et al., 2019), which is specifically developed to score clinical text by capturing a variety of ways clinical concepts can be expressed. INCITE is a rule-based modular pipeline utilizing custom-built lexicons, which contain observed misspellings for medical concepts and non-standard expressions, as well as common concepts and abbreviations from online resources. The tool performs direct and fuzzy matching between a new response and an annotated response (or a lexicon variant of it) using a fixed or dynamic Levenshtein ratio threshold (in our case - .95). Full details about the INCITE system are available in Sarker et al. (2019).

**Learning curves:** We compare the approaches by experimenting with different training set sizes and evaluating on the same test set of 20% held-out data (1,5K responses for Set 1 and 2,5K for Set 2). This provides insight on an important practical consideration - how much training data is enough

to train a reliable and accurate model (Heilman and Madnani, 2015). To emulate an operational scenario, the division of training and test sets (and the increase in training data) are based on the chronological order in which the responses were received.

**Evaluation metrics** Another practical consideration is to directly answer two questions of operational significance: "How accurate is the system for responses that it is able to score?" and "How many responses do human raters still need to score manually?". To address these, we present two separate metrics – *F1* for matched responses and *total number of unmatched responses* – as opposed to capturing the number of unmatched responses through the measure of Recall. This setup allows the selection of more strict or liberal thresholds depending on the intended use, e.g., high-stakes summative assessment where high precision is paramount vs. formative assessment, where there can be a trade off between precision and wider response coverage.

**Thresholds:** A conservative similarity threshold of .95 is selected apriori to ensure high confidence that the matched responses are scored correctly. All items below that threshold are considered unmatched and are sent for human scoring. We first present detailed results for this threshold. Next, we experiment with a variety of other thresholds and compare their effect on the two evaluation metrics.

#### 5 Results

The majority class baseline was .79 for Set 1 and .794 for Set 2. The remaining results for a threshold of .95 are presented in Table 1. As can be seen, all three systems (INCITE, ACTA No finetuning, and ACTA Finetuned) achieve very high F1 scores for the responses they were able to match for Set 1 (lowest F1 was .977 for ACTA Finetuned and .984 for INCITE). For the much larger Set 2, we see a higher F1 score range of .97 - .99 for ACTA

<sup>4</sup><https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>

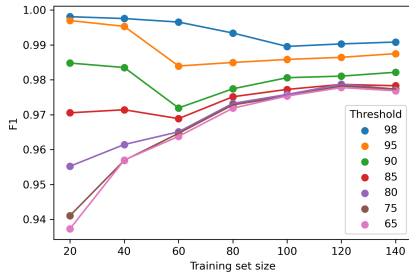


Figure 1: F1 score for Set 1 (SHARP items) as a function of similarity threshold and training set size.

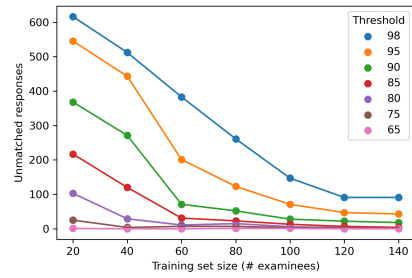


Figure 3: Number of unmatched responses for Set 1 (SHARP items) as a function of similarity threshold and training set size.

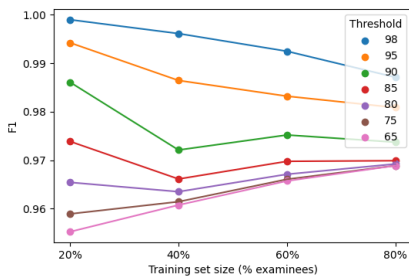


Figure 2: F1 score for Set 2 (SAQs) as a function of similarity threshold and training set size.

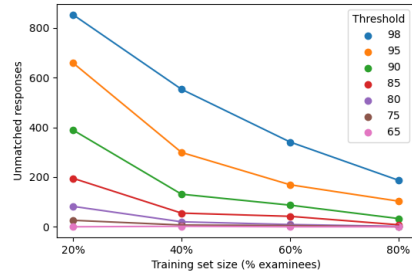


Figure 4: Number of unmatched responses for Set 2 (SAQ items) as a function of similarity threshold and training set size.

compared to .88 - .90 for INCITE. The F1 score remains high when evaluation is performed using 5-fold cross validation (not shown in the tables): the average ACTA Finetuned F1 across folds for Set 1 is .985 with an average number of unmatched responses across folds = 49.8. For Set 2 the F1 score is .98 with an average number of unmatched responses across folds = 88.8. Overall, the results suggest a consistently high level of confidence in ACTA’s output for all matched responses.

When looking at the *unmatched* responses, we see dramatic differences between the three systems. When training on more than 40 examinees, *INCITE* and *ACTA No finetuning* have significantly more responses that require human review and increasing the amount of training data leads to small improvements. *ACTA Finetuned* leaves fewer unmatched responses and continuously improves with the addition of more training data. These results show the when finetuned using contrastive loss, ACTA can ultimately save more human effort than INCITE and that the gains increase with data size.

Next, we experiment with different matching thresholds by replacing the .95 value with a range of values: .98, .90, .85, .80, .75, .70, and .65. F1 remains high even with lower thresholds: For Set 1, the lowest F1 is .937 (threshold = .65 when training

on data from 20 examinees). For Set 2 it is .95 for the same configuration (for detailed F1 results for each threshold, see figures 1 and 2). The number of unmatched responses, however, decreases significantly (see Figures 3 and 4) – there are either 0 or 1 unmatched responses in both sets across all training configurations for threshold .65. This shows that with more liberal thresholds, the need for human scoring almost disappears (except the need for continuous quality verification). Selecting the right trade-off between F1 and number of responses that need to undergo human review remains an operational decision.

## 6 Conclusion

This study showed that a similarity-based clinical ASAG system finetuned using contrastive loss outperforms the INCITE and ACTA No Finetuning baselines. Lowering the similarity threshold value significantly decreases the number of unmatched responses, while – contrary to expectation – the F1 score remains high at > .93 across conditions. The condition of weakest supervision – training on 20 examinees from Set 1 with a similarity threshold of .65 – shows that 880 annotated responses are

sufficient to score *all* 1.5K test set responses with  $F1 = .93$ . Similarly, when training on 20% of the data from Set 2 with threshold of  $.65$ , *all* 2.5K test set responses are scored with  $F1 = .95$ .

The evaluation setup allows operational experts to balance the confidence threshold with a minimum necessary F1 score, where items with more errors can have more stringent similarity thresholds and vice-versa. The threshold may also vary depending on intended use: formative exams may tolerate a lower F1 to gain wider coverage, while summative assessments may have stricter criteria.

In addition to its accuracy and wider coverage of responses, the interpretability of ACTA as a similarity-based system is an important advancement in clinical assessment compared to instance-based ASAG systems (e.g., Ha et al. (2020)). Interpretability holds special significance in the realm of automated scoring, as the value of the scores depends on the trust placed by various stakeholders (such as faculty, students, and residency selection programs, among others) in their fairness, reliability, and validity.

Like many other products, automated scoring tools are complex systems that have a significant impact not only because of their technical capabilities but also due to how they are used and the way their results are interpreted. Misusing these tools or interpreting their outputs incorrectly can lead to serious ethical issues. In a summative context, the models described in this article are intended to be used as hybrid systems, where human raters always review borderline cases. In a formative context, it is crucial to carefully examine the relationship between the use of the system and its impact on learning outcomes, as this is essential evidence for validity.

Next steps include exploration of the effects of different "gaming" strategies (e.g., intentionally providing generic instead of specific answers) and potential differential functioning across demographic groups. Notably, ACTA is intended as a hybrid system, where cases of examinees who perform near or below the passing standard are reviewed by human experts.

## References

Marie Bexte, Andrea Horbach, and Torsten Zesch. 2022. Similarity-based content scoring-how to make s-bert keep up with bert. In *Proceedings of the 17th Work-*

*shop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 118–123.

Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE.

Myroslava O Dzikovska, Rodney D Nielsen, Chris Brew, Claudia Leacock, Danilo Giampiccolo, Luisa Bentivogli, Peter Clark, Ido Dagan, and Hoa T Dang. 2013. Semeval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge. Technical report, NORTH TEXAS STATE UNIV DENTON.

Le Ha, Victoria Yaneva, Polina Harik, Ravi Pandian, Amy Morales, and Brian Clauser. 2020. Automated prediction of examinee proficiency from short-answer questions.

Michael Heilman and Nitin Madnani. 2015. The impact of training data on automated short answer scoring performance. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 81–85.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Keisuke Sakaguchi, Michael Heilman, and Nitin Madnani. 2015. Effective feature integration for automated short answer scoring. In *Proceedings of the 2015 conference of the North American Chapter of the association for computational linguistics: Human language technologies*, pages 1049–1054.

Abeed Sarker, Ari Z Klein, Janet Mee, Polina Harik, and Graciela Gonzalez-Hernandez. 2019. An interpretable natural language processing system for written medical examination assessment. *Journal of biomedical informatics*, 98:103268.

Johannes Schneider, Robin Richner, and Micha Riser. 2022. Towards trustworthy autograding of short, multi-lingual, multi-type answers. *International Journal of Artificial Intelligence in Education*, pages 1–31.

## A Appendix

```
batch_size = 32; log_every_n_step = 100;
lr = 0.00002; margin = 0.5; max_length
= 512; model_name_or_path = "sentence-
transformers/all-MiniLM-L6-v2"; num_epochs =
1; num_training_participants = 142; num_workers
= 8; threshold = 0.95; warmup_ratio = 0.1;
weight_decay = 0.01
```

# Hybrid Models for Sentence Readability Assessment

Fengkai Liu, John S. Y. Lee

Department of Linguistics and Translation

City University of Hong Kong

Hong Kong SAR, China

fengkaliu3-c@my.cityu.edu.hk, jsylee@cityu.edu.hk

## Abstract

Automatic readability assessment (ARA) predicts how difficult it is for the reader to understand a text. While ARA has traditionally been performed at the passage level, there has been increasing interest in ARA at the sentence level, given its applications in downstream tasks such as text simplification and language exercise generation. Recent research has suggested the effectiveness of hybrid approaches for ARA, but they have yet to be applied on the sentence level. We present the first study that compares neural and hybrid models for sentence-level ARA. We conducted experiments on graded sentences from the Wall Street Journal (WSJ) and a dataset derived from the OneStopEnglish corpus. Experimental results show that both neural and hybrid models outperform traditional classifiers trained on linguistic features. Hybrid models obtained the best accuracy on both datasets, surpassing the previous best result reported on the WSJ dataset by almost 13% absolute.

## 1 Introduction

Text readability is defined as the cognitive load of a reader to comprehend a text (Martinc et al., 2021). Research on automatic readability assessment (ARA) has traditionally aimed at passages (Azziazu and Pera, 2019), e.g., labeling a passage with its difficulty level.

There has been growing interest in assessing the difficulty of individual sentences (Štajner et al., 2017; Brunato et al., 2018; Lu et al., 2020; Schicchi et al., 2020), given its application in various downstream tasks in natural language processing (NLP). It is essential to generation tasks that are sensitive to language difficulty, such as pedagogical material and exercises (Pilán et al., 2014). It also facilitates explainable text simplification (Gârbacea et al., 2021) by identifying which sentences require simplification. Sentence-level ARA is a task in its own right since a substantial drop in performance

has been observed when passage-level ARA models are applied on individual sentences (Kilgarriff et al., 2008; Pilán et al., 2016).

Similar to many other NLP tasks, passage-level ARA has benefited from the advent of neural approaches (Filighera et al., 2019; Tseng et al., 2019; Martinc et al., 2021). Recent research has also applied ‘hybrid’ models, which leverage both linguistically motivated features and neural models (Deutsch et al., 2020; Lee et al., 2021; Lim et al., 2022). For sentence-level ARA, although neural models have been evaluated (Schicchi et al., 2020; Arase et al., 2022), there has not been any attempt to integrate linguistic features.

This paper applies neural models and hybrid models on sentence-level ARA and compares their performance with a non-neural classifier trained on linguistic features. To our knowledge, this is the first study on hybrid models for sentence-level ARA. Experimental results show that a hybrid model offers the best performance, and surpasses the previous best result reported on the Wall Street Journal dataset (Brunato et al., 2018).<sup>1</sup>

## 2 Previous work

### 2.1 Neural and hybrid approaches

Readability formulas (Kincaid et al., 1975) and traditional approaches for readability assessment have mostly relied on one-hot linguistic features and language models (Collins-Thompson, 2008; Sung et al., 2015). More recent studies have shown that neural approaches can improve assessment performance (Azziazu and Pera, 2019; Martinc et al., 2021). An active area of ARA research is to investigate how to incorporate linguistic features into neural models. On passage-level assessment, some studies observed no effect (Deutsch et al., 2020) or only marginal improvement (Filighera et al., 2019)

<sup>1</sup>All data and code are publicly released at <https://github.com/fliu6/Hybrid4SentenceARA>.



from linguistic features, while others reported significant improvement, e.g. by combining Random Forest and RoBERTa (Lee et al., 2021), and concatenating linguistic features with sentence embeddings from BERT hidden layers (Imperial, 2021). However, there has not yet been any study on hybrid models on sentence-level ARA.

## 2.2 Sentence readability assessment

Most previous research on sentence readability pursued binary classification or pairwise difficulty prediction (Ambati et al., 2016; Schumacher et al., 2016). An algorithm combining rule-based and statistical classifiers yielded 71% accuracy on binary classification of texts for learning Swedish as a foreign language (Pilán et al., 2014). Statistical classifiers achieved 66% accuracy on an English dataset based on Wikipedia and Simple Wikipedia (Vajjala and Meurers, 2014) and between 78.9% and 83.7% on an Italian dataset (Dell’Orletta et al., 2014).

There have also been a few studies on sentence-level ARA involving multi-way classifiers trained with traditional machine learning methods. Brunato et al. (2018) developed an SVM linear regression model with a variety of surface, morphological and syntactic features. The model achieved 59.1% and 60% accuracy on an Italian and an English dataset of sentences graded on a 7-point scale. Sentence length and nominal modification were found to correlate significantly with sentence difficulty. A Bayesian Ridge Regression Model, trained on a variety of linguistic features including syntax, lexical, morphology and cohesion, has been shown to achieve high correlation with human judgment on German sentence difficulty (Weiss and Meurers, 2022). A classifier has also been trained on features derived from the phrase complexity level of n-grams (Štajner et al., 2017). It attained 0.66 weighted F-score on an English dataset on a 5-point scale. A classifier for Chinese sentences, based on vocabulary and grammar points, reached 31.92% accuracy on 10-way classification (Lu et al., 2020).

Two studies have applied neural models on sentence-level ARA. Schicchi et al. (2020) showed that an RNN-based architecture outperformed Vec2Read (Mikolov et al., 2013). Arase et al. (2022) found that the BERT-base model outperformed traditional machine learning classifiers on their annotated CEFR-based sentence difficulty dataset. However, they did not attempt to incorporate any linguistic features. This paper aims to

fill in this gap with a comparison of neural models, hybrid models and traditional classifiers.

## 3 Data

We used the following two datasets in our experiments. Detailed statistics are shown in Table 3 and Table 4 (see Appendix A).

### 3.1 Wall Street Journal (WSJ)

This corpus (Brunato et al., 2018) consists of 1,200 sentences drawn from the Wall Street Journal (Nivre et al., 2007) and graded on a difficult scale from 1 to 7. Each sentence was rated by 20 native speakers on a difficult scale from 1 (“very easy”) to 7 (“very difficult”). Our evaluation is based on the set of 650 sentences whose grade was agreed upon by at least 14 of the 20 annotators. While it is possible to restrict the evaluation to sentences with an even higher rate of agreement, it would lead to a substantially smaller dataset, whose size is already much smaller than other datasets.<sup>2</sup>

### 3.2 OneStopEnglish (OSE)

This corpus (Vajjala and Lučić, 2018) consists of aligned texts graded at three reading grades: beginner, intermediate, and advanced. Each of the 189 texts has three versions corresponding to these grades, with a total of 19,904 sentences in the 567 texts.<sup>3</sup>

Instead of assigning the grade of the text to all sentences in that text (Pilán et al., 2014), we determined the difficulty of each individual sentence based on the human revision. Among the sentences in intermediate texts, 10.21% appear verbatim in the beginner version; among those in the advanced texts, 18.76% appear verbatim in one of the lower versions. These sentences are labeled with the lowest grade at which they appear. All other sentences are labeled with the grade of the text — the fact the human editors revised them implies that their grade could not be lower.

## 4 Approach

### 4.1 Baseline: Linguistic Model

We used the scikit-learn implementation of Random Forests (RF) and XGBoost (XGB) (Pedregosa

<sup>2</sup>No sentence in this subset was graded at 6 or 7.

<sup>3</sup>Sentence segmentation was performed with NLTK (Bird et al., 2009).

et al., 2011). We extracted 255 linguistic features with LingFeat<sup>4</sup> for each sentence. We performed feature selection with the Variance Threshold in scikit-learn on the dev set.<sup>5</sup> Similar to Lu et al. (2020), we trained these classifiers with linguistic features as well as bag-of-word features.

## 4.2 Baseline: Neural Model

Transformer-based neural models have achieved impressive performance in many natural language processing tasks.

We fine-tuned BERT (Devlin et al., 2019), BART (Lewis et al., 2019), RoBERTa (Liu et al., 2019), XLNet (Yang et al., 2019) and ELECTRA (Clark et al., 2020) on our datasets (Section 3) into an ARA classifier<sup>6</sup>, using the pre-trained versions released by Huggingface (Wolf et al., 2019). We used the base versions of all of the above, as well as the large versions of BART, RoBERTa and ELECTRA.

## 4.3 Hybrid Models

We implemented three hybrid models. The following model incorporates linguistic features into a neural model:

**Concatenated Model** Similar to Song et al. (2021), the input to model consists of the input sentence  $w_1w_2\dots w_n$  concatenated with the linguistic features  $f_1, f_2\dots f_n$ , in the format “[CLS]  $w_1w_2\dots w_n$  [SEP]  $f_1f_2\dots f_n$ ”.

The following two models wrap the linguistic features and neural model output in a non-neural statistical classifier:

**Hard Label** Following Deutsch et al. (2020), the grade of the sentence, as predicted by the Neural Model (Section 4.2), serves as an additional feature in the statistical classifier (Section 4.1).

**Soft Labels** Following Lee et al. (2021), the probability of each grade, as predicted by the Neural Model (Section 4.2), serve as additional features alongside the linguistic features in the statistical classifier (Section 4.1).

<sup>4</sup><https://github.com/brucewlee/lingfeat>

<sup>5</sup>The threshold set to 0.8.

<sup>6</sup>We used the Adam algorithm (Kingma and Ba, 2015) for optimization. The epoch for each training is 10, and set the maximum word embedding size as 128.

# 5 Experiments

## 5.1 Set-up

We report results in terms of accuracy (Acc.), F1-score, Precision, Recall and QWK scores.

We used stratified ten-fold cross validation in WSJ and OSE experiments, with a 8:1:1 split for training, development and testing.<sup>7</sup> For the OSE dataset, all sentences from the same text are placed in the same fold, so that the entities and topics mentioned in the test sentences would not be seen during training.

## 5.2 Results

**Linguistic Model.** XGBoost (XGB) outperformed Random Forest (RF) and Linear Regression (LR) on all datasets. On OSE and WSJ, it achieved 0.451 and 0.618 accuracy, respectively, compared to 0.412 and 0.551 for RF, and 0.374 and 0.413 for LR. We will therefore present results based on XGB in the remainder of this section.

**Neural Model.** Table 1 presents the performance of neural models on the WSJ and OSE datasets. On the WSJ dataset, RoBERTa obtained the best performance among base versions, at a 0.668 accuracy. Large models were found to outperform base versions on the WSJ dataset, in which BART-large produced the highest accuracy at 0.679. On the OSE dataset, BART obtained the best performance among base versions, at a 0.571 accuracy. Large models were also found to outperform base versions on the OSE dataset, in which BART-large produced the highest accuracy at 0.571. Generally, BART-large model achieved the best performance on all datasets, at 0.679 and 0.571 accuracy for the WSJ and OSE datasets, respectively. We will therefore use its predictions for hybrid models.

The results for OSE and WSJ in Table 2 are based on the BART-large model, which obtained the best performance on both datasets. Consistent with results from passage-level ARA, the Neural Model achieved better performance over the Linguistic Model on both datasets in all metrics. Despite the relatively small amount of training data in the WSJ datasets, the Neural Model still offered competitive performance.

**Hybrid Models.** The previous best published result for the WSJ dataset 0.600, obtained with an

<sup>7</sup>The hyperparameters for learning rate, dropout and batch size are tuned on the dev set. We found best performance with learning rate at  $1 \cdot e^{-5}$ , dropout at 0.2, and set batch size as 32.

Dataset	Metric	BERT base	BART base	RoBERTa base	XLNet base	ELECTRA base	BART large	RoBERTa large	ELECTRA large
WSJ	Acc.	0.606	0.648	0.668	0.640	0.602	<b>0.679</b>	0.667	0.630
	F1	0.527	0.590	0.596	0.540	0.520	<b>0.611</b>	0.603	0.523
	Prec.	0.480	0.566	0.576	0.469	0.477	<b>0.601</b>	0.589	0.453
	Recall	0.606	0.648	0.668	0.640	0.602	<b>0.679</b>	0.667	0.630
	QWK	0.540	0.678	0.640	0.601	0.552	0.661	<b>0.677</b>	0.552
OSE	Acc.	0.547	0.571	0.569	0.562	0.555	<b>0.571</b>	0.570	0.566
	F1	0.532	0.555	0.554	0.543	0.533	<b>0.558</b>	0.555	0.549
	Prec.	0.549	0.570	0.566	0.554	0.552	0.565	<b>0.567</b>	0.566
	Recall	0.547	0.571	0.569	0.562	0.555	<b>0.571</b>	0.570	0.566
	QWK	0.500	0.537	0.537	0.535	0.512	<b>0.549</b>	0.541	0.532

Table 1: ARA performance of the Neural Model based on different transformers

Dataset	Metric	Linguistic Model	Neural Model	Hybrid Model		
				Concatenated	Hard Label	Soft Labels
WSJ	Acc.	0.618	0.679	0.629	<b>0.729</b>	0.724
	F1	0.549	0.611	0.590	0.707	<b>0.709</b>
	Prec.	0.519	0.601	0.585	0.713	<b>0.715</b>
	Recall	0.618	0.679	0.629	<b>0.729</b>	0.724
	QWK	0.616	0.661	0.676	0.767	<b>0.794</b>
OSE	Acc.	0.451	0.571	0.568	0.578	<b>0.581</b>
	F1	0.428	0.558	0.559	<b>0.565</b>	0.564
	Prec.	0.441	0.565	0.584	<b>0.593</b>	0.574
	Recall	0.451	0.571	0.568	0.578	<b>0.581</b>
	QWK	0.288	0.549	0.540	0.537	<b>0.560</b>

Table 2: ARA performance of the Linguistic Model, Neural Model (BART-large) and Hybrid Model

SVM model (Brunato et al., 2018). The Hybrid Model with Hard Label surpassed this result by almost 13% absolute to achieve state-of-the-art result, at 0.729 accuracy. The Soft Labels Model produced the second best performance, followed by the Neural Model. The Concatenated Model did not outperform the Neural Model, which may be because long complex sequences and the size of dataset easily lead to overfit on the transformer-based models. The improvement of the Hard Label Model over the Neural Model<sup>8</sup> was statistically significant.

On the OSE dataset, the Soft Labels Model obtained the best performance in accuracy, though at a lower accuracy (0.581) than on the WSJ dataset. This likely reflects more fuzzy boundaries between the categories in the OSE corpus, where all sentences in the original texts were used. The Hard Label Model produced the second best performance as OSE dataset, followed by the Neural Model also. The Concatenated Model obtained worse perfor-

mance than Neural Model also. The improvement of the Soft Label Model over the Neural Model<sup>9</sup> was statistically significant.

## 6 Conclusion

We have presented the first study on hybrid models on automatic readability assessment (ARA) at the sentence level. Our contribution is two-fold. First, we demonstrated that hybrid models outperform neural models, suggesting that linguistic features can capture salient properties that indicate sentence difficulty. Second, we compared three types of hybrid model, and showed that using the neural model’s predictions as features in a traditional classifier yielded the best result, surpassing the previous best published result on the WSJ dataset by almost 13% absolute. These experimental results are expected to help inform future research on sentence-level ARA.

<sup>8</sup>At  $p < 3.6 \cdot e^{-6}$  according to McNemar’s Test.

<sup>9</sup>At  $p < 1.4 \cdot e^{-4}$  according to McNemar’s Test.

## 7 Limitation

Our experimental results should be interpreted with the following limitations in mind. First, our experiments involved relatively small datasets in English only. The performance of the model should also be evaluated on other languages and larger datasets. Second, the improvement observed in our best models depends on both the efficacy of the linguistic features and on the strength of the neural model itself. As neural models continue to improve and effective linguistic features are identified, the best methods for combining may also need to be updated.

## Acknowledgement

This work was partly supported by the Language Fund from the Standing Committee on Language Education and Research (project EDB(LE)/PR/EL/203/14) and by the General Research Fund (project 11207320).

## References

- Bharat Ram Ambati, Siva Reddy, and Mark Steedman. 2016. Assessing Relative Sentence Complexity using an Incremental CCG Parser. In *Proceedings of NAACL-HLT 2016*.
- Yuki Arase, Satoru Uchida, and Tomoyuki Kajiwara. 2022. *CEFR-based sentence difficulty annotation and assessment*. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6206–6219, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Ion Madrazo Azpiazu and Maria Soledad Pera. 2019. Multiattentive recurrent neural network architecture for multilingual readability assessment. *Transactions of the Association for Computational Linguistics*, 7:421–436.
- Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc."
- Dominique Brunato, Lorenzo De Mattei, Felice Dell'Orletta, Benedetta Iavarone, and Giulia Venturi. 2018. Is this sentence difficult? do you agree? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2690–2699.
- Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.
- Kevyn Collins-Thompson. 2008. Computational assessment of text readability: A survey of current and future research. *International Journal of Applied Linguistics*, 165(2):97–135.
- Felice Dell'Orletta, Martijn Wieling, Andrea Cimino, Giulia Venturi, and Simonetta Montemagni. 2014. Assessing the Readability of Sentences: Which Corpora and Features? In *Proc. 9th Ninth Workshop on Innovative Use of NLP for Building Educational Applications*.
- Tovly Deutsch, Masoud Jasbi, and Stuart Shieber. 2020. Linguistic Features for Readability Assessment. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pretraining of Deep Bidirectional Transformers for Language Understanding. In *Proc. North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT)*.
- Anna Filighera, Tim Steuer, and Christoph Rensing. 2019. Automatic text difficulty estimation using embeddings and neural networks. In *European Conference on Technology Enhanced Learning*, page 335–348. Springer.
- Cristina Gârbacea, Mengtian Guo, Samuel Carton, and Qiaozhu Mei. 2021. Explainable Prediction of Text Complexity: The Missing Preliminaries for Text Simplification. In *Proc. 59th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Joseph Marvin Imperial. 2021. Bert embeddings for automatic readability assessment. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 611–618.
- Adam Kilgarriff, Mils Husák, Katy McAdam, Michael Rundell, and Pavel Rychlý. 2008. GDEX: Automatically Finding Good Dictionary Examples in a Corpus. In *Proc. EURALEX*.
- Peter J. Kincaid, Robert P. Fishburne, Richard L. Rogers, and Brad S. Chissom. 1975. Derivation of new readability formulas for Navy enlisted personnel. In *Research Branch Report 8-75*. Chief of Naval Technical Training: Naval Air Station Memphis.
- Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proc. 3rd International Conference for Learning Representations*, San Diego.
- Bruce W. Lee, Yoo Sung Jang, and Jason Hyung-Jong Lee. 2021. Pushing on Text Readability Assessment: A Transformer Meets Handcrafted Linguistic Features. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

- 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*.
- Ho Hung Lim, Tianyuan Cai, John S. Y. Lee, and Meichun Liu. 2022. Robustness of hybrid models in cross-domain readability assessment. In *Proceedings of the The 20th Annual Workshop of the Australasian Language Technology Association*, pages 62–67, Adelaide, Australia. Australasian Language Technology Association.
- 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*.
- Dawei Lu, Xinying Qiu, and Yi Cai. 2020. Sentence-level readability assessment for l2 chinese learning. *CLSW 2019, LNAI*, 11831:381–392.
- Matej Martinc, Senja Pollak, Marko, and Robnik-Šikonja. 2021. Supervised and Unsupervised Neural Approaches to Text Readability. *Computational Linguistics*, 47(1):141–179.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proc. International Conference on Learning Representations (ICLR)*.
- Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The conll 2007 shared task on dependency parsing. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 915–932.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, and O. Grisel. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.
- Ildikó Pilán, Sowmya Vajjala, and Elena Volodina. 2016. A Readable Read: Automatic Assessment of Language Learning Materials based on Linguistic Complexity. *International Journal of Computational Linguistics and Applications*, 7(1):143–159.
- Ildikó Pilán, Elena Volodina, and Richard Johansson. 2014. Rule-based and Machine Learning Approaches for Second Language Sentence-level Readability. In *Proc. 9th Workshop on Innovative Use of NLP for Building Educational Applications*.
- Daniele Schicchi, Giovanni Pilato, and Giosué Lo Bosco. 2020. Deep neural attention-based model for the evaluation of italian sentences complexity. In *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*, pages 253–256. IEEE.
- Elliot Schumacher, Maxine Eskenazi, Gwen Frishkoff, and Kevyn Collins-Thompson. 2016. Predicting the relative difficulty of single sentences with and without surrounding context. In *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1871–1881.
- Dandan Song, Siyi Ma, Zhanchen Sun, Sicheng Yang, and Lejian Liao. 2021. Kvl-bert: Knowledge enhanced visual-and-linguistic bert for visual commonsense reasoning. *Knowledge-Based Systems*, 230:107408.
- Yao-Ting Sung, Ju-Ling Chen, Ji-Her Cha, Hou-Chiang Tseng, Tao-Hsing Chang, and Kuo-En Chang. 2015. Constructing and validating readability models: the method of integrating multilevel linguistic features with machine learning. *Behavior Research Methods*, 47:340–354.
- Hou-Chiang Tseng, Hsueh-Chih Chen, Kuo-En Chang, Yao-Ting Sung, and Berlin Chen. 2019. An Innovative BERT-Based Readability Model. In *Lecture Notes in Computer Science, vol 11937*.
- Sowmya Vajjala and Ivana Lučić. 2018. Onestopenglish corpus: A new corpus for automatic readability assessment and text simplification. In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 297–304.
- Sowmya Vajjala and Detmar Meurers. 2014. Assessing the relative reading level of sentence pairs for text simplification. In *Proc. 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, page 288–297.
- Sanja Štajner, Simone Paolo Ponzetto, and Heiner Stuckenschmidt. 2017. Automatic Assessment of Absolute Sentence Complexity. In *Proc. 26th International Joint Conference on Artificial Intelligence (IJCAI)*.
- Zarah Weiss and Detmar Meurers. 2022. Assessing sentence readability for German language learners with broad linguistic modeling or readability formulas: When do linguistic insights make a difference? In *Proc. 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*, pages 141 – 153.
- 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. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

## A Appendix: Corpus statistics

WSJ		
Score	# sent	sent length
1	69	10.43
2	262	14.51
3	203	25.00
4	96	30.70
5	20	31.50
Total	650	20.27

Table 3: Size of the WSJ dataset and the average sentence length

OSE		
Version	# sent	sent length
Beginner	4,840	18.75
Intermediate	4,759	22.44
Advanced	4,632	25.90
Total	14,231	22.31

Table 4: Size of the OSE dataset and the average sentence length

# Training for Grammatical Error Correction Without Human-Annotated L2 Learners' Corpora

Mikio Oda

National Institute of Technology, Kurume College, Japan

oda@kurume-nct.ac.jp

## Abstract

Grammatical error correction (GEC) is a challenging task for non-native second language (L2) learners and learning machines. Data-driven GEC learning requires as much human-annotated genuine training data as possible. However, it is difficult to produce larger-scale human-annotated data, and synthetically generated large-scale parallel training data is valuable for GEC systems. In this paper, we propose a method for rebuilding a corpus of synthetic parallel data using target sentences predicted by a GEC model to improve performance. Experimental results show that our proposed pre-training outperforms that on the original synthetic datasets. Moreover, it is also shown that our proposed training without human-annotated L2 learners' corpora is as practical as conventional full pipeline training with both synthetic datasets and L2 learners' corpora in terms of accuracy.

## 1 Introduction

Grammatical error correction (GEC) is one of the essential processes needed to produce sentences in a grammar-based language, and it is a challenging task for non-native second language (L2) learners and learning machines as well. Each language has its own grammar, however, data-driven language learning by a machine does not use the grammar, but corpora, more preferably, large-scale corpora. While classifiers that predict some token from candidates for a certain position in a sentence have been developed in the past (Li et al., 2019), sequence-to-sequence models have become more popular for GEC because the task is regarded as a sequence-to-sequence one and the models are flexible in editing sentences and covering various error types.

In sequence-to-sequence models, Felice et al. (2014) and Junczys-Dowmunt and Grundkiewicz (2014) treat the task as a statistical machine translation (SMT) problem and produce state-of-the-art

performance on the CoNLL2014 shared task. Neural machine translation models (Sutskever et al., 2014), which consist of an encoder and a decoder, also have been investigated to improve their capabilities. In particular, the Transformer (Vaswani et al., 2017), which is an encoder-decoder model incorporating a self-attention mechanism, has become popular and various improved versions have been investigated. One of its alternative architectures is the Copy-Augmented Transformer, which has become popular for GEC (Hotate et al., 2020).

Another modification to the Transformer architecture is altering the encoder-decoder attention mechanism in the decoder to accept and make use of additional context. For example, Kaneko et al. (2019) use the BERT representation of the input sentence as additional context for GEC. GECToR (Omelianchuk et al., 2020) employs a BERT-like pre-trained encoder stacked with a linear layer with the softmax activation function, and treats the GEC task as a token labeling problem. Addressing training data for GEC models, Kiyono et al. (2019), Grundkiewicz et al. (2019) and Choe et al. (2019) employ synthetically generated pseudo data for pre-training of GEC systems prior to fine-tuning on human-annotated corpora for the Building Educational Applications (BEA) 2019 shared task (Bryant et al., 2019).

This paper addresses the effectiveness of synthetic parallel data, which is generally used as a consequence of the insufficiency of human-annotated L2 learners' corpora. We propose a method of substituting target sentences in synthetic parallel data with alternatives and rebuilding synthetic datasets to boost GEC training. Experiments demonstrate that pre-training on synthetic datasets rebuilt by the proposed method outperforms pre-training on the original synthetic datasets. Moreover, our synthetic datasets can be effectively employed not only to pre-train, but also to fine-tune GEC models, that is, training on synthetic data only

all through the pipeline. The GEC model’s training without L2 learners’ corpora is as practical as conventional training with both synthetic datasets and L2 learners’ corpora in terms of accuracy.

## 2 Synthetic parallel training data

### 2.1 Generating synthetic training data

Supervised machine learning requires as much genuine training data as possible, and the same is true for GEC. Training data or corpora for GEC may be created with annotations by trained native speakers of the language or by grammarians. This fact makes it difficult for us to produce larger-scale genuine data, so researchers are compelled to use limited resources to train their learning models (Bryant et al., 2019). Therefore, synthetically generated large-scale parallel training data contributes to GEC systems along with the human-annotated data.

Synthetic parallel training data consist of erroneous sentences generated by corruption models from error-free sentences. In general, the corruption models can generate unlimited versions of erroneous sentences from a given error-free one, with the ability to vary the versions in the number of errors, error types, etc. Back-translation (Sennrich et al., 2016) provides monolingual training data with synthetic source sentences that are obtained from automatically translating the target sentence into the source language for NMT. Kiyono et al. (2019) apply back-translation to GEC and achieves state-of-the-art performance on the CoNLL2014 and BEA2019 test datasets.

PIE synthetic data (Awasthi et al., 2019) is often used in state-of-the-art GEC models proposed by Omelianchuk et al. (2020); Sorokin (2022), etc. Seq2Edits (Stahlberg and Kumar, 2020) is a sequence-to-sequence transducer which consists of a Transformer encoder and decoders, and can predict span-based edit operation probabilities for GEC. Stahlberg and Kumar (2021), furthermore, propose tagged corruption models using both Seq2Edits and a finite state transducer to match the observed error type distribution of the BEA2019 dev dataset, and generate synthetic data for pre-training GEC models.

### 2.2 Problems in synthetic training data

Given some noise to an error-free (grammatically correct) sentence, a system can generate a different version of the sentence which is generally regarded

as a grammatically incorrect sentence. However, it does not always become an incorrect sentence. Table 1 shows some examples of inappropriate edits on the PIE-9M<sup>1</sup> and the C4-200M<sup>2</sup> synthetic datasets. The PIE model (Awasthi et al., 2019) and the tagged corruption model (Stahlberg and Kumar, 2021) each applies deletion to the source sentence, removing an adverb. In the PIE-9M synthetic dataset, the system removes the word *also* from the source sentence  $y^1$  to generate the erroneous sentence (Corrupted), and the edit to correct the sentence is *missing also* to recover from the error. However, the removed word is not necessarily required for the sentence  $x^1$  because it is an additive adverb, so the corrupted sentence  $x^1$  itself is an error-free sentence whose edit should be *no-operation*. The table also shows the same case in the C4-200M synthetic dataset. Note that *Source* is a target sentence to be outputted from a GEC model and *Corrupted* is a source sentence inputted to the model. The examples are cases where the original error-free sentences (Source) are inappropriate for the target sentences.

Large-scale synthetic parallel training datasets are often used to pre-train a GEC model prior to its fine-tuning on small-scale genuine datasets. The genuine datasets for the fine-tuning are annotated by trained native speakers of the language with respect to L2 learners’ mistakes because the GEC model is expected to correct L2 learners’ mistakes in text. Synthetic data for pre-training, therefore, should also match the data characteristics of L2 learners’ grammatical mistakes as shown in human-annotated datasets to be employed in the final training. The corruption mechanism produces unexpected inappropriate edits on synthetic data that differ from human errors. Finally, synthetic data, itself, is one of the key resources for building better GEC systems.

## 3 Erroneous synthetic data rebuilt by GEC models

In this section, we further examine the problem described in the previous section and propose to rebuild conventional synthetic datasets, which are often employed by researchers, in order to create effective synthetic parallel training datasets for pre-training. A trained GEC model can be

<sup>1</sup><https://github.com/awasthiabhijeet/PIE/>

<sup>2</sup>[https://github.com/google-research-datasets/C4\\_200M-synthetic-dataset-for-grammatical-error-correction/](https://github.com/google-research-datasets/C4_200M-synthetic-dataset-for-grammatical-error-correction/)



PIE 9M	
Source $\mathbf{y}^1$ :	There have <b>also</b> been recent battles over access to multiple myeloma drug lenolidamide.
Corrupted $\mathbf{x}^1$ :	There have been recent battles to access to multiple myeloma drug lenolidamide.
Predicted $\tilde{\mathbf{y}}^1$ :	There have been recent battles to access to multiple myeloma drug lenolidamide.
C4-200M	
Source $\mathbf{y}^2$ :	We <b>just</b> have to live with black that are not truly black.
Corrupted $\mathbf{x}^2$ :	We have to live in black that are not black.
Predicted $\tilde{\mathbf{y}}^2$ :	We have to live in black that are not black.

Table 1: Examples of inappropriate edits of synthetic data for GEC. *Source* is an error-free sentence that is treated as a target sentence in a GEC training model. *Corrupted* is regarded as a grammatically incorrect sentence that is treated as a source sentence in the model. *Predicted* is a hypothetical target sentence generated by a GEC model. The bold words are not in the corrupted sentences; however, these words are not missing words that make the sentences ungrammatical.

represented by  $g(\mathbf{x}^i)$ , where  $\mathbf{x}^i (= (x_1^i, \dots, x_n^i))$  is the  $i$ th erroneous input sentence with tokens  $x_j^i (1 \leq j \leq n)$ ,  $g(\mathbf{x}^i)$  is the  $i$ th predicted output sentence:  $\tilde{\mathbf{y}}^i = (\tilde{y}_1^i, \dots, \tilde{y}_m^i)$ . We train the model  $g$  with given datasets of incorrect and correct sentence pairs:  $\mathcal{D} = \{(\mathbf{x}^i, \mathbf{y}^i) | i = 1, \dots, N\}$ , where the size of  $\mathcal{D}$  is  $N$ , so as to decrease the difference (loss) of  $\mathbf{y}^i$  between  $\tilde{\mathbf{y}}^i$ .

### 3.1 Process of generating synthetic data

Fig.1 shows a general process for generating synthetic parallel data consisting of an incorrect and correct sentence pair. The sentence  $\mathbf{y}^i$  is an error-free sentence from a large-scale corpus such as Wikipedia, BookCorpus (Zhu et al., 2015) and the Colossal Clean Crawled Corpus (C4) (Raffel et al., 2020), and a corruption model produces some grammatical errors in the sentence  $\mathbf{y}^i$  resulting in an erroneous sentence  $\mathbf{x}^i$ . The sentences  $\mathbf{x}^i$  and  $\mathbf{y}^i$  are the input sentence to a GEC model and the sentence that should be inferred by the model, respectively. The arrow from  $\mathbf{y}^i$  to  $\mathbf{x}^i$  is a noising process to add the errors, and the reverse dotted arrow is a de-noising process to restore the erroneous sentence to the correct form. In some cases, however, the target sentence of the noisy or erroneous sentence  $\mathbf{x}^i$  should not be the unedited sentence  $\mathbf{y}^i$ , but another sentence  $\hat{\mathbf{y}}^i$ .

The noising and de-noising processes of the corruption models, therefore, often have irreversibility, and the hypothetically correct sentence  $\hat{\mathbf{y}}^i$  does not always match the unedited error-free sentence  $\mathbf{y}^i$ . On the other hand, the process of generating a correct sentence  $\hat{\mathbf{y}}^i$  from the erroneous sentence  $\mathbf{x}^i$  by human annotators on genuine parallel data matches the correction process, and can create a

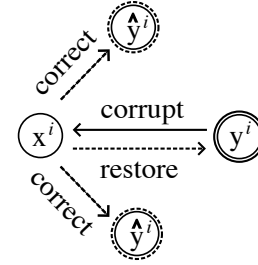


Figure 1: Process of generating a synthetic sentence pair  $(\mathbf{x}^i, \mathbf{y}^i)$ . The sentence  $\mathbf{y}^i$  is an error-free sentence from a large-scale corpus. The sentence  $\mathbf{x}^i$  is an erroneous sentence generated by a corruption model.

dataset  $\hat{\mathcal{D}} = \{(\mathbf{x}^i, \hat{\mathbf{y}}^i) | i = 1, \dots, N\}$  which is significantly reliable as long as the annotators do not make mistakes. Even in human-annotated data, there can be plural candidates for the correct sentence  $\hat{\mathbf{y}}^i$ , but, the dataset  $\hat{\mathcal{D}}$  is still reliable (Bryant et al., 2019).

### 3.2 Proposed method for rebuilding synthetic data

We address synthetic data for GEC models and propose a modification where hypothetical target sentences are not original unedited sentences  $\mathbf{y}^i$ , but sentences predicted from corrupted ones by a conventional GEC model. In other words, we rebuild the synthetic data  $\tilde{\mathcal{D}} = \{(\mathbf{x}^i, \tilde{\mathbf{y}}^i) | i = 1, \dots, N\}$  from  $\mathcal{D} = \{(\mathbf{x}^i, \mathbf{y}^i) | i = 1, \dots, N\}$  which are usually used in pre-training of GEC models. This idea is similar to Rothe et al. (2021).

We employ a conventional GEC model  $g(\mathbf{x}^i)$  to generate hypothetical target sentences  $\tilde{\mathbf{y}}^i$ . One would expect that the predicted sentences  $\tilde{\mathbf{y}}^i$  from corrupted sentences  $\mathbf{x}^i$  by a GEC model would match the corrected sentences  $\hat{\mathbf{y}}^i$ :  $\tilde{\mathbf{y}}^i \simeq \hat{\mathbf{y}}^i$  for  $\mathbf{x}^i$ ,

and build an appropriate synthetic dataset :  $\tilde{\mathcal{D}} \simeq \hat{\mathcal{D}}$ . The conventional GEC model we employ in this paper is GECToR (Omelianchuk et al., 2020), where the number of labels is 5,004. GECToR has achieved state-of-the-art results on GEC, however, the version of the model we employ achieves  $F_{0.5}$  scores of 64.0 and 71.8 on the CoNLL 2014 and BEA 2019 test datasets, respectively. As the GEC systems, of course, are still under development by researchers, we have to compromise on the quality of synthetic data rebuilt by our proposed method. Table 1 also shows examples of hypothetical target sentences  $\tilde{\mathbf{y}}^i$ , which contain grammatical errors, generated by the GEC model.

### 3.3 Synthetic data rebuilt by the GEC model

To predict  $\tilde{\mathbf{y}}^i$  from  $\mathbf{x}^i$  we employ a newer version of the trained GECToR model<sup>3</sup> which has a RoBERTa encoder based on the results of Omelianchuk et al. (2020) and the inference hyperparameters, *confidence bias* and *minimum probability* threshold, are set to 0.2 and 0.5, respectively. As synthetic data to be examined, we use the above-mentioned PIE-9M and C4-200M in the experiments; the former is widely used for pre-training GEC models and the latter is generated by attempting to match the error type frequency distribution to the development dataset. Note that the C4-200M dataset is downsized to 9M sentences to match the size of the PIE-9M in the experiments.

Table 2 shows the fundamental statistics of the synthetic datasets rebuilt by the proposed method, compared to the original ones. The average numbers of tokens per sentence in the rebuilt datasets  $\tilde{\mathcal{D}}$ s are not significantly different from those of the original datasets  $\mathcal{D}$ s. To compare statistical relationships between sentences  $\mathbf{x}^i$  and  $\mathbf{y}^i$ , we generate m2 formatted information using the ERROR ANnotation Toolkit (ERRANT)<sup>4</sup>(Bryant et al., 2017) and calculate the average number of edits per sentence. Applying the proposed method to the PIE-9M and C4-200M train datasets, the procedure reduces the average number of edits (corruptions) per sentence, resulting in about 0.8 and 2.8 fewer than the original datasets, respectively. We also indicate the dataset  $\tilde{\mathcal{D}}$ , which has a comparable average number of edits with the dataset  $\tilde{\mathcal{D}}$ . The erroneous sentences  $\tilde{\mathbf{x}}^i$  are generated from the corrupted sentences  $\mathbf{x}^i$  in the PIE-9M dataset by recovering edits

partly to adjust its average of edits to that of the dataset  $\tilde{\mathcal{D}}$ . The dataset  $\tilde{\mathcal{D}}$  is used in the experiments in the next section to prove that the effectiveness of our method does not depend on the number of edits per sentence empirically.

Stahlberg and Kumar (2021) have tried to match their synthetic data characteristics to L2 learners' error characteristics with respect to the frequency of occurrence of the error types for the reason that the trained model is mainly expected to correct L2 learners' sentences. We further examine whether our method can regulate the frequency of occurrence with respect to grammatical error types in the synthetic datasets to match the L2 learners'. Fig.2 shows the frequency distribution of occurrence with respect to grammatical error types in our rebuilt synthetic datasets  $\tilde{\mathcal{D}}$ s, comparing the original synthetic datasets  $\mathcal{D}$ s, PIE-9M and C4-200M, and L2 learners' corpus, the Cambridge English Write & Improve (W&I+LOCNESS) v2.1<sup>5</sup>(Bryant et al., 2019; Granger, 1998). The proposed method changes the frequency of error occurrence, and we expect that the frequency distribution of  $\mathcal{D}$  could approach that of the L2 learners' corpus by the proposed method. Note that the L2 learners' corpus for comparison is employed in stage III training of GEC models, which is the final fine-tuning stage in the experiments, and the corpus for the final stage of training is of utmost importance.

To investigate the similarity between two frequency distributions, we calculate Kullback-Leibler (KL) divergence, which is a measure of how different two probability distributions are from each other, defined as

$$D_{\text{KL}}(P||Q) = \sum_{x \in \chi} P(x) \log \left( \frac{P(x)}{Q(x)} \right), \quad (1)$$

where  $P$  and  $Q$  are discrete probability distributions and  $\chi$  is the sample space. We consider the frequency distributions as the probability distributions, and the sample space  $\chi$  is 24 error types defined by ERRANT. Table 2 also shows the average level of information, i.e., entropy. The entropy measures uncertainty of the types of grammatical errors that will occur in a sentence.

Comparing each entropy value of the proposed synthetic datasets  $\tilde{\mathcal{D}}$ s with that of their original ones  $\mathcal{D}$ s, the proposed method approaches the entropy of the PIE-9M synthetic data and that of the W&I LOCNESS dataset  $\mathcal{D}_{WI}$ , while there is no

<sup>3</sup><https://github.com/grammarly/gector/>

<sup>4</sup><https://github.com/chrisjbryant/errant/>

<sup>5</sup><https://www.cl.cam.ac.uk/research/nl/bea2019st/>

Synthetic (#sentences)	Dataset	$\mathbf{x}^i/\mathbf{y}^i/D$	#tokens	#edits	Entropy [bit]	$D_{\text{KL}}(\mathcal{D}_{\text{WI}}  \cdot)$	$D_{\text{KL}}(\mathcal{D}_{\text{Co}}  \cdot)$
PIE-9M (8.42M)		$\mathbf{x}^i$	25.1	—	—	—	—
		$\tilde{\mathbf{x}}^i$	25.2	—	—	—	—
		$\mathbf{y}^i$	25.4	—	—	—	—
		$\hat{\mathbf{y}}^i$	25.1	—	—	—	—
	Original	$\mathcal{D}(\mathbf{x}^i, \mathbf{y}^i)$	—	2.45	3.79	0.216	<b>0.198</b>
	Proposed	$\tilde{\mathcal{D}}(\mathbf{x}^i, \tilde{\mathbf{y}}^i)$	—	1.60	3.87	<b>0.186</b>	0.216
Random	$\check{\mathcal{D}}(\tilde{\mathbf{x}}^i, \mathbf{y}^i)$	—	1.62	3.79	0.198	0.216	
C4-200M (8.42M)		$\mathbf{x}^i$	25.7	—	—	—	—
		$\mathbf{y}^i$	25.7	—	—	—	—
		$\hat{\mathbf{y}}^i$	25.8	—	—	—	—
	Original	$\mathcal{D}(\mathbf{x}^i, \mathbf{y}^i)$	—	4.04	3.80	<b>0.093</b>	<b>0.177</b>
Proposed	$\tilde{\mathcal{D}}(\mathbf{x}^i, \tilde{\mathbf{y}}^i)$	—	1.26	3.80	0.196	0.369	
W&I+LOC		$\mathcal{D}_{\text{WI}}(\mathbf{x}^i, \mathbf{y}^i)$	—	—	3.88	—	0.128
CoNLL2014		$\mathcal{D}_{\text{Co}}(\mathbf{x}^i, \mathbf{y}^i)$	—	—	3.86	0.143	—

Table 2: The statistical metrics of the sentences and the datasets, where  $\mathbf{x}^i$  are corrupted sentences from the corresponding error-free sentences  $\mathbf{y}^i$  in the original corpus. The proposed method creates hypothetical correct sentences  $\tilde{\mathbf{y}}^i$  of the dataset  $\tilde{\mathcal{D}}$ . The comparative partially recovered sentences  $\tilde{\mathbf{x}}^i$  are created to match the average number of edits per sentence to the proposed  $\tilde{\mathcal{D}}$ . W&I+LOC is its train dataset and CoNLL2014 is its test dataset.

significant difference from the C4-200M dataset. In the PIE-9M synthetic dataset, the proposed method also approaches the frequency distribution of the types of grammatical errors to that of  $\mathcal{D}_{\text{WI}}$ . Regarding the C4-200M dataset, on the other hand, the proposed method moves the frequency distribution away from that of  $\mathcal{D}_{\text{WI}}$ , however, the two datasets rebuilt by the proposed method,  $\tilde{\mathcal{D}}$ s, have almost the same value of KL divergence from  $\mathcal{D}_{\text{WI}}$ . The table also refers to the values of KL divergence from the CoNLL2014 dataset for evaluating the GEC models. Note that the CoNLL2014 dataset is small-sized and consists of 1,312 sentences.

## 4 Experiments

To empirically investigate the effectiveness of the proposed method and the capabilities of a GEC model trained on synthetic data rebuilt by the method, we train the GEC model choosing the hyperparameters described below. The GEC model is fundamentally trained through the three stage pipeline adopted in Choe et al. (2019), Omelianchuk et al. (2020), Stahlberg and Kumar (2021), etc.: stage I is a pre-training stage on a synthetic dataset, stage II is a training stage on a human-annotated dataset and stage III is a fine-tuning stage on a smaller human-annotated dataset more consistent with the target domain of GEC.

### 4.1 Training model and datasets

In the experiments, we employ RoBERTa (Liu et al., 2019)(roberta-base<sup>6</sup>) and train the model on the datasets indicated below. Hyperparameters in the training stage are set to the same values as on the website<sup>7</sup> (Omelianchuk et al., 2020), and choosing a set of labels to be predicted by the model is done in the same manner as described there. We also employ three different PIE-9M and three different C-200M datasets.

**Stage I (Pre-training)** Either the PIE-9M or the C-200M is used in stage I as a conventional method. Each dataset consists of 9M sentence pairs, which we randomly split into two sets: 95% train and 5% dev datasets. The data splitting creates 8.42M sentence-pair synthetic parallel datasets  $\mathcal{D}$ s. We apply the proposed method to the above datasets  $\mathcal{D}$ s to create the proposed synthetic parallel datasets  $\tilde{\mathcal{D}}$ s. We also create the dataset  $\check{\mathcal{D}}$  which has a similar average number of edits per sentence by recovering some edits randomly and partially to adjust to the statistics of the proposed datasets. The statistical information for all the synthetic parallel datasets is shown in Table 2. Note that all text in the C-200M dataset is tokenized using spaCy and the en\_core\_web\_sm model<sup>8</sup>.

<sup>6</sup><https://huggingface.co/models/>

<sup>7</sup><https://github.com/grammarly/gector/>

<sup>8</sup><https://spacy.io/>

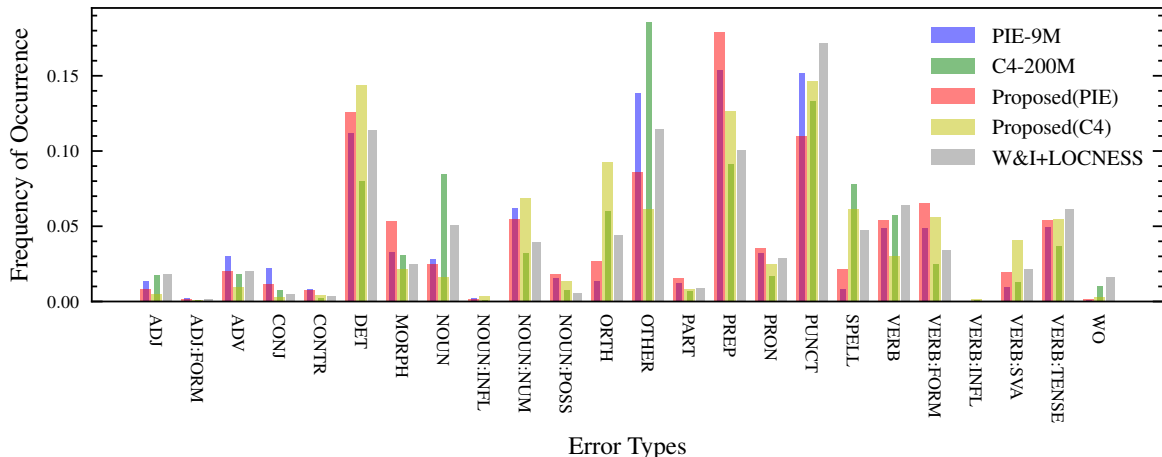


Figure 2: The frequency distribution of occurrence with respect to grammatical error types defined by ERRANT. ERRANT analyzes grammatical errors, which are categorized into 24 types, in sentences  $x^i$  by comparing those to target sentences  $y^i$  or  $\tilde{y}^i$ . The statistics of the synthetic datasets rebuilt by the proposed method are compared with the original synthetic datasets and L2 learners’ corpus.

**Stage II (Training)** We employ L2 learners’ human-annotated corpora used in the BEA2019 shared task. The corpora consist of W&I+LOCNESS v2.1, the First Certificate in English (FCE) v.2.1 (Yannakoudakis et al., 2011), the National University of Singapore Corpus of Learner English (NUCLE) (Dahlmeier et al., 2013) and the Lang-8 Corpus of Learner English (Lang-8) (Mizumoto et al., 2011; Tajiri et al., 2012) shown in Table 3. We split the corpora into 98% train and 2% dev datasets because they are small-sized and train data of a larger size is preferable. Table 3 shows the characteristics of each corpus and the overall corpus for stages II and III.

**Stage III (Fine-tuning)** We choose W&I+LOCNESS, one of the corpora in stage II, as an L2 learners’ corpus consistent with the target domain of GEC. This selection is based on Choe et al. (2019) for the restricted track and Omelianchuk et al. (2020). In addition to the L2 learners’ corpus, the synthetic dataset rebuilt by the proposed method is downsized to 34K sentence pairs for fine-tuning of the models pre-trained on the same synthetic data. The sentence pairs of the downsized synthetic dataset are chosen randomly from the 9M sentence pairs.

## 4.2 Results

We trained the GEC models on either of the original PIE-9M, C4-200M or our rebuilt synthetic datasets in stage I followed by training in a combination of stages II and III. Both stages II and III use the

Dataset	Stage	#sents	#tokens	#edits
W&I+L A	II, III	10,490	17.5	2.69
W&I+L B	II, III	13,030	18.3	1.83
W&I+L C	II, III	10,781	19.2	0.926
FEC	II	28,345	16.0	1.52
NUCLE	II	56,957	20.3	0.758
Lang-8	II	1.04M	11.4	1.20
total(train)	II	1.13M	12.2	1.24
total(train)	III	33,614	18.3	1.84

Table 3: L2 learners’ corpora employed in training stages II and/or III. The number of sentence pairs, the average number of tokens and edits per sentence are indicated for each corpus. Each corpus is split into train and dev datasets, and the overall train data for stages II and III is also shown. Note that *sentence* means a token sequence to be inputted to the model, and each sentence in the W&I+LOCNESS is assigned to a CEFR level, A, B or C.

L2 learners’ corpora or our rebuilt 34K synthetic datasets described in Sec. 4.1. To evaluate the performance of the trained models, we let each model correct grammatical errors in the sentences of the CoNLL2014 and BEA2019 test datasets. Note that we set the *confidence bias* and the *minimum probability* threshold to zeros for inference after stages I and II as on the website. We evaluated the performance of the models for the CoNLL2014 and BEA2019 test datasets using M<sup>2</sup>scorer<sup>9</sup> and by submitting the corrected sentences to the server

<sup>9</sup><https://github.com/nusnlp/m2scorer/>

referred to by the BEA2019 shared task website<sup>10</sup>, respectively.

Table 4 shows comparisons of GEC performance with metrics, precision (P), recall (R) and  $F_{0.5}$  scores for the test datasets, indicating train datasets each model used in stages I, II and III. The results for the PIE-9M synthetic dataset are summarized as follows. The baselines are the underlined results of the model trained on the conventional datasets, that is, *Original+BEA2019* through stages I, II and III, resulting in  $F_{0.5} = 62.9$  and  $F_{0.5} = 70.5$  for the CoNLL2014 and BEA2019 test datasets, respectively. While the pre-trained *Original* performs  $F_{0.5} = 51.2$  and  $F_{0.5} = 51.1$ , the pre-trained *Proposed* performs  $F_{0.5} = 61.2$  and  $F_{0.5} = 66.7$ , respectively. For the partial pipeline training of stages I and III, the *Original+BEA2019* performs  $F_{0.5} = 62.4$  and  $F_{0.5} = 70.3$ , and the *Proposed+BEA2019* performs  $F_{0.5} = 62.8$  and  $F_{0.5} = 70.1$ , respectively. *Proposed+PIE-34K*, which was pre-trained and fine-tuned only on the rebuilt PIE-9M and PIE-34K synthetic datasets, performs  $F_{0.5} = 62.9$  and  $F_{0.5} = 71.5$ , respectively. *Proposed+C4-34K* was pre-trained and fine-tuned only on the synthetic datasets as well, however, the training employed two different synthetic datasets, PIE and C4. For the full pipeline training of stages I, II and III, the *Original+BEA2019* performs  $F_{0.5} = 62.9$  and  $F_{0.5} = 70.5$ , and the *Proposed* performs  $F_{0.5} = 63.6$  and  $F_{0.5} = 70.6$ , respectively. The results regarding the C4-200M synthetic dataset are also shown in the same manner in the figure.

## 5 Discussion and related work

This paper addresses the *quality* of synthetic parallel data due to the insufficiency of human-annotated L2 learners' corpora and the effectiveness of training only on synthetic data. Note that the *quality* does not address grammatical correctness, but the validity of source-target sentence pairs for training and how well the data fits the characteristics of L2 learners' mistakes. The overall results indicate that our method is more effective for the PIE-9M dataset than the C4-200M dataset, and it implies that the C4-200M dataset is of better quality.

Here, we discuss the experiments on the PIE-9M dataset, which more likely needs the technique. The stage-I training by the proposed method outperforms the conventional training by 10.0 and

15.6 with regard to  $F_{0.5}$  for the CoNLL2014 and BEA2019 test datasets, respectively. It results in only 1.7 and 3.8 less than the baselines, which were trained through the full pipeline, stages I, II, and III.

Furthermore, the stage-II training reduces the performance of the pre-trained models on the proposed method's synthetic dataset. This suggests that the proposed method's synthetic datasets could be of higher quality than the overall L2 learners' corpora while each synthetic dataset itself could be inferior to the L2 learners' corpora.

Unfortunately, the baseline of the training replaced with the rebuilt synthetic dataset does not improve its performance. Our synthetic datasets can be employed all through the pipeline of training, that is, training without L2 learners' corpora. The results show that GEC model training without L2 learners' corpora is as practical as conventional training with both L2 learners' corpora and synthetic datasets in terms of accuracy. Note that the version of the model employed to rebuild synthetic data in the experiments achieves the scores of 64.0 and 71.8 on  $F_{0.5}$  for the CoNLL 2014 and BEA 2019 test datasets, respectively.

To summarize the achievements, the proposed method :

1. outperforms pre-training on the original synthetic datasets.
2. provides notably good training performance without human-annotated L2 learners' corpora.

Trained GEC models can be used not only for predicting correct sentences but also for generating better synthetic data, and systems incorporating the proposed method are not limited to the synthetic data and model used in this paper.

Addressing training data for GEC models, Grundkiewicz and Junczys-Dowmunt (2014) introduce the WikEd Error Corpus generated from Wikipedia revision histories, corpus content and format. The corpus consists of more than 12 million sentences with a total of 14 million edits of various types. Kiyono et al. (2019), Grundkiewicz et al. (2019) and Choe et al. (2019) employ synthetically generated pseudo data for pre-training of GEC systems prior to fine-tuning on human-annotated corpora for the BEA2019 shared task (Bryant et al., 2019).

<sup>10</sup><https://www.cl.cam.ac.uk/research/nl/bea2019st/>

Synthetic	Training Datasets	Stage			CoNLL2014 test			BEA2019 test		
		I	II	III	P	R	F <sub>0.5</sub>	P	R	F <sub>0.5</sub>
PIE-9M	Original	<i>S</i>			60.4	31.8	51.2	54.3	41.5	51.1
	PartiallyRecovered	<i>S</i>			58.7	32.3	50.4	51.2	42.9	49.3
	Proposed	<i>S</i>			66.5	46.3	<b>61.2</b>	68.3	60.9	<b>66.7</b>
	Original+BEA2019	<i>S</i>	<i>A</i>		64.2	45.3	<b>59.2</b>	61.3	58.5	<b>60.7</b>
	PartiallyRecovered+BEA2019	<i>S</i>	<i>A</i>		63.7	45.5	59.0	60.0	59.3	59.8
	Proposed+BEA2019	<i>S</i>	<i>A</i>		64.0	45.1	59.1	60.0	58.8	59.8
	Original+BEA2019	<i>S</i>		<i>A</i>	72.4	40.2	62.4	76.4	53.4	70.3
	PartiallyRecovered+BEA2019	<i>S</i>		<i>A</i>	72.8	39.1	62.1	76.4	52.3	70.0
	Proposed+BEA2019	<i>S</i>		<i>A</i>	70.7	43.5	62.8	74.3	57.0	70.1
	<b>Proposed+PIE-34K</b>	<i>S</i>		<i>S</i>	73.9	39.5	<b>62.9</b>	78.1	53.5	71.5
	<b>Proposed+C4-34K</b>	<i>S</i>		<i>S</i>	75.1	37.5	62.5	79.7	52.2	<b>72.1</b>
	Original+BEA2019	<i>S</i>	<i>A</i>	<i>A</i>	69.1	46.8	<u>62.9</u>	75.0	56.6	<u>70.5</u>
	PartiallyRecovered+BEA2019	<i>S</i>	<i>A</i>	<i>A</i>	73.3	42.1	<b>63.9</b>	75.0	56.5	70.4
	Proposed+BEA2019	<i>S</i>	<i>A</i>	<i>A</i>	73.4	41.5	63.6	75.8	55.3	<b>70.6</b>
C4-200M	Original	<i>S</i>			64.2	39.1	56.9	62.9	50.4	59.9
	Proposed	<i>S</i>			66.3	47.9	<b>61.6</b>	68.1	62.0	<b>66.8</b>
	Original+BEA2019	<i>S</i>	<i>A</i>		65.6	46.3	<b>60.6</b>	61.2	60.5	<b>61.0</b>
	Proposed+BEA2019	<i>S</i>	<i>A</i>		63.7	45.9	59.1	59.6	59.5	59.5
	Original+BEA2019	<i>S</i>		<i>A</i>	72.5	42.1	63.3	78.1	56.3	<b>72.5</b>
	Proposed+BEA2019	<i>S</i>		<i>A</i>	70.9	44.7	63.4	73.1	58.8	69.7
	<b>Proposed+C4-34K</b>	<i>S</i>		<i>S</i>	75.3	40.0	<b>64.0</b>	77.9	54.6	71.8
	<b>Proposed+PIE-34K</b>	<i>S</i>		<i>S</i>	74.8	39.9	63.6	78.2	54.3	71.8
	Original+BEA2019	<i>S</i>	<i>A</i>	<i>A</i>	72.9	43.1	<b>64.0</b>	75.8	58.3	<b>71.5</b>
	Proposed+BEA2019	<i>S</i>	<i>A</i>	<i>A</i>	73.4	41.4	63.6	75.1	56.3	70.4

Table 4: Comparison of GEC performance after pre-training (stage I) on either the original synthetic datasets or the datasets rebuilt by the proposed method. The pre-trained models were further trained on either the L2-learners’ corpora or the rebuilt synthetic datasets in stages II and/or III. *S* and *A* mean Synthetic and Annotated train datasets, respectively.

Mita et al. (2020) focus on human annotators’ errors in official datasets when they rewrite incorrect sentences to remove grammatical mistakes and denoise the target sentences of the official datasets using some trained GEC models with a perplexity criterion. Rothe et al. (2021) also apply the similar technique to the LANG-8 corpus, which is a large corpus of texts written by L2 learners with user-annotated corrections, and correct human errors by the GEC models.

Our proposed method is effective not only for correcting human annotators’ errors, but also for adjusting source-target disparity to match the domain. Stahlberg and Kumar (2021) build a large synthetic pre-training dataset with error tag frequency distributions matching Seq2Edits (Stahlberg and Kumar, 2020). Parnow et al. (2021) trained a generator to generate increasingly realistic errors (in the form of token-based edit labels) and a discrimina-

tor to differentiate between artificially-generated edits and human-annotated edits. Stahlberg and Kumar (2021) propose tagged corruption models using both the Seq2Edits and a finite state transducer to match the observed error type distribution of the BEA2019 dev dataset, and generate synthetic data for pre-training GEC models. Yasunaga et al. (2021) apply BIFI algorithm (Yasunaga and Liang, 2021) and LM-Critic to synthetic data to generate better datasets for GEC. LM-Critic chooses the most likely grammatical sentence from multiple sentence candidates based on the sentence occurrence probabilities generated by a language model.

## 6 Conclusion

In this paper, we have addressed the effectiveness of synthetic parallel data and have proposed a method for rebuilding a corpus of synthetic parallel data using target sentences predicted by a GEC

model. While the original target sentences in synthetic parallel data are guaranteed to be error-free, the target sentences predicted by a GEC model contain grammatical errors because the GEC model has been developed through research and is not perfect in its performance. However, pre-training on our proposed synthetic data outperforms that on the original synthetic data, and our pre-trained GEC model showed performance only slightly lower than the conventional fine-tuned GEC model. In addition, our proposed method can provide notably good training performance without human-annotated L2 learners' corpora.

The proposed method's target sentences by an imperfect GEC model work better than the original error-free target sentences although the former may contain grammatical errors. The reason why this paradoxical result happens needs to be determined. In future work, we plan to investigate further re-configuration and modification of synthetic parallel data, and fine-tune training using such data to improve the performance of GEC. Investigation of the source-target relationships on training data mentioned above should also be carried out to clarify the effectiveness of the proposed method.

## Acknowledgements

We gratefully thank Martin Chodorow at CUNY Hunter College for his valuable suggestions and feedback. Furthermore, we would like to thank the reviewers for their insightful comments. This work was supported by JSPS KAKENHI (Grant Numbers JP18K00904 and JP21K00806).

## References

- Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. [Parallel iterative edit models for local sequence transduction](#). 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 4260–4270, Hong Kong, China. Association for Computational Linguistics.
- Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. [The BEA-2019 shared task on grammatical error correction](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.
- Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. [Automatic annotation and evaluation of error types for grammatical error correction](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.
- Yo Joong Choe, Jiyeon Ham, Kyubyong Park, and Yeoil Yoon. 2019. [A neural grammatical error correction system built on better pre-training and sequential transfer learning](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 213–227, Florence, Italy. Association for Computational Linguistics.
- Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. [Building a large annotated corpus of learner English: The NUS corpus of learner English](#). In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31, Atlanta, Georgia. Association for Computational Linguistics.
- Mariano Felice, Zheng Yuan, Øistein E. Andersen, Helen Yannakoudakis, and Ekaterina Kochmar. 2014. [Grammatical error correction using hybrid systems and type filtering](#). In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 15–24, Baltimore, Maryland. Association for Computational Linguistics.
- Sylviane Granger. 1998. *The computer learner corpus: A versatile new source of data for SLA research*. Addison Wesley Longman, London and New York.
- Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2014. [The WikEd error corpus: A corpus of corrective Wikipedia edits and its application to grammatical error correction](#). In *Advances in Natural Language Processing*, pages 478–490, Cham. Springer International Publishing.
- Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. [Neural grammatical error correction systems with unsupervised pre-training on synthetic data](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263, Florence, Italy. Association for Computational Linguistics.
- Kengo Hotate, Masahiro Kaneko, and Mamoru Komachi. 2020. [Generating diverse corrections with local beam search for grammatical error correction](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2132–2137, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2014. [The AMU system in the CoNLL-2014 shared task: Grammatical error correction by data-intensive and feature-rich statistical machine translation](#). In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 25–33, Baltimore, Maryland. Association for Computational Linguistics.

- Masahiro Kaneko, Kengo Hotate, Satoru Katsumata, and Mamoru Komachi. 2019. [TMU transformer system using BERT for re-ranking at BEA 2019 grammatical error correction on restricted track](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 207–212, Florence, Italy. Association for Computational Linguistics.
- Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. [An empirical study of incorporating pseudo data into grammatical error correction](#). 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 1236–1242, Hong Kong, China. Association for Computational Linguistics.
- Ruobing Li, Chuan Wang, Yefei Zha, Yonghong Yu, Shiman Guo, Qiang Wang, Yang Liu, and Hui Lin. 2019. [The LAIX systems in the BEA-2019 GEC shared task](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 159–167, Florence, Italy. 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. 2019. [RoBERTa: A robustly optimized BERT pretraining approach](#). *arXiv preprint arXiv:1907.11692*.
- Masato Mita, Shun Kiyono, Masahiro Kaneko, Jun Suzuki, and Kentaro Inui. 2020. [A self-refinement strategy for noise reduction in grammatical error correction](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 267–280, Online. Association for Computational Linguistics.
- Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2011. [Mining revision log of language learning SNS for automated Japanese error correction of second language learners](#). In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 147–155, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.
- Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzshanskiy. 2020. [GECToR – grammatical error correction: Tag, not rewrite](#). In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online. Association for Computational Linguistics.
- Kevin Parnow, Zuchao Li, and Hai Zhao. 2021. [Grammatical error correction as GAN-like sequence labeling](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3284–3290, Online. Association for Computational Linguistics.
- 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.
- Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. [A simple recipe for multilingual grammatical error correction](#). 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 702–707, 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.
- Alexey Sorokin. 2022. [Improved grammatical error correction by ranking elementary edits](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11416–11429, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Felix Stahlberg and Shankar Kumar. 2020. [Seq2Edits: Sequence transduction using span-level edit operations](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5147–5159, Online. Association for Computational Linguistics.
- Felix Stahlberg and Shankar Kumar. 2021. [Synthetic data generation for grammatical error correction with tagged corruption models](#). In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 37–47, Online. Association for Computational Linguistics.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. [Sequence to sequence learning with neural networks](#). In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS’14*, page 3104–3112, Cambridge, MA, USA. MIT Press.
- Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. [Tense and aspect error correction for ESL learners using global context](#). In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 198–202, Jeju Island, Korea. 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.



- Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. [A new dataset and method for automatically grading ESOL texts](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.
- Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2021. [LM-critic: Language models for unsupervised grammatical error correction](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7752–7763, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Michihiro Yasunaga and Percy Liang. 2021. [Break-it-fix-it: Unsupervised learning for program repair](#). In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11941–11952. PMLR.
- Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urta-sun, A. Torralba, and S. Fidler. 2015. [Aligning books and movies: Towards story-like visual explanations by watching movies and reading books](#). In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27, Los Alamitos, CA, USA. IEEE Computer Society.

# Exploring a New Grammatico-functional Type of Measure as Part of a Language Learning Expert System

Cyriel Mallart<sup>1</sup>, Andrew Simpkin<sup>2</sup>, Rémi Venant<sup>3</sup>, Nicolas Ballier<sup>4</sup>,  
Bernardo Stearns<sup>5</sup>, Jen Yu Li<sup>1</sup>, Thomas Gaillat<sup>1</sup>

<sup>1</sup>LIDILE, Université Rennes 2

<sup>2</sup>School of Mathematics, Statistics and Applied Mathematics, University of Galway

<sup>3</sup>LIUM, Université du Mans

<sup>4</sup>CLILLAC-ARP, Université Paris Cité

<sup>5</sup>Insight, Data Science Institute, University of Galway

## Abstract

This paper explores the use of L2-specific grammatical microsystems as elements of the domain knowledge of an Intelligent Computer-assisted Language Learning (ICALL) system. We report on the design of new grammatico-functional measures and their association with proficiency. We illustrate the approach with the design of the IT, THIS, THAT proform microsystem. The measures rely on the paradigmatic relations between words of the same linguistic functions. They are operationalised with one frequency-based and two probabilistic methods, i.e., the relative proportions of the forms and their likelihood of occurrence. Ordinal regression models show that the measures are significant in terms of association with CEFR levels, paving the way for their introduction in a specific proform microsystem expert model.

## 1 Introduction

This paper explores the use of L2-specific grammatical systems as elements of the domain knowledge of an Intelligent Computer-assisted Language Learning (ICALL) system. Such systems rely on Natural Language Processing approaches that conduct several high-end tasks such as Grammatical Error Detection (GED), automatic reformulation or proficiency level prediction. As part of the Intelligent Tutoring System (ITS) category, they rely on models that have an expertise, which is language use in their case.

Expert models encapsulate the domain knowledge which is required to describe the learner's language skills involved in tasks such as writing production. In ITSs, there are several possible strategies used to acquire and represent domain knowledge (Nkambou et al., 2010). Among

those are rule-based cognitive models, describing learning strategies, and Constraint-based models (CBM) describing principles that rely on correct solutions to a problem.

In the case of ICALL, representing the knowledge of learners has traditionally been done within the Constraint-based model (CBM) framework thanks to correct-usage principles derived from native language use. For instance, some Grammatical Error Detection tasks are processed on the basis of target hypotheses (TH) (Lüdeling and Hirschmann, 2015), i.e. the correct version of what is meant by a learner in a specific segment. In this type of tasks, correct versions of erroneous segments or patterns are compared with the TH (Bryant et al., 2017) to identify incorrect uses. As useful as it has proved to be, this type of approach tends to reduce the knowledge about L2 language production to what native speakers would say or write by focusing on error correction. In doing so, it overlooks the meta-knowledge that language learning experts possess regarding acquisition processes. Experts' evaluations of learner language not only rely on TH, but also on what they know of the grammatical, lexical, semantic and pragmatic features in L2 writings of different proficiency levels, be they negative or positive features (Bulté and Housen, 2012). This allows them to position the learner's productions in terms of level and to provide feedback.

We argue that an expert ICALL system should not be reduced to error identification and correction on the basis of native language production, but include comprehensive knowledge about the range of L2 linguistic profiles at different stages of language learning. We intend to use such profiling as part of a learning-analytics system providing in-

formation to teachers on their learners' linguistic developmental stages.

Modelling the domain knowledge with such profiles linked to proficiency levels is necessary. In order to do so, we draw inspiration from rule-based cognitive models. The role of rule-based cognitive models is to describe the knowledge involved in "student performance in a given task domain, including strategies, problem-solving principles, and knowledge of how to apply problem-solving principles in the context of specific problems" (Alevén, 2010). When applied to language learning, this approach complies well with describing the strategies used to elaborate language patterns, including idiosyncrasies. This makes rule-based cognitive models quite comprehensive in describing learner language characteristics.

Our proposal follows this approach, as it considers an expert model as a cognitive entity that knows positive and negative characteristics of an L2 set of writings at various stages of proficiency. The expert model should not simply "know" the rules that operate for native speakers, it should also include the probability of patterns that govern specific levels. Many grammatico-functional microsystems (MS) exist that describe a part of the grammatical reasoning at work in production. They are convenient to describe the psychological reality of the learner and may be linked to proficiency as in the English Grammar Profile (O'Keeffe and Mark, 2017).

As an illustration of the broad process, we present the design and implementation of a specific linguistic microsystem as a rule-based cognitive model, namely the THIS, THAT and IT proform microsystem. Our working hypothesis revolves around the idea that different proficiency levels prompt different linguistic contexts around the use of the microsystem, which leads to different odds of using the forms in the microsystem. Therefore, by observing the probability of using a given form in the microsystem as a function of the context, we could capture aspects of the learner's grammatical reasoning that points to a given proficiency level. This approach raises two research questions:

1. What is the likelihood of microsystem forms in L2 writings, according to the linguistic context that surrounds the microsystem?
2. What is the distribution of these probabilities across CEFR levels?

To answer these questions, we propose to use a model to describe the probabilities of use of THIS, THAT and IT proforms depending on context, as a first step toward modelling linguistic profiles. In other words, this model predicts the likelihood of a learner using either THIS, THAT or IT given the linguistic context of this proform, while being agnostic to whether the choice of such proform was correct or not. To assess the relevance of using the likelihood of microsystem forms as linguistic profiling, a second model predicts proficiency levels using only the probabilities of using THIS, THAT and IT output by the microsystem prediction model. If this second classification model can discriminate proficiency levels using only the predicted probabilities for the forms of the microsystem, then the microsystem likelihood model is a coherent way to build domain knowledge indicators for profiling. In section 2 we present the theoretical background underlying our research. Section 3 presents the data and the microsystem extraction methods used to exploit it. In section 4, we present how the microsystems are implemented and evaluated in terms of extraction and predictability. Section 5 covers the results obtained with different modelling approaches to validate the associations between microsystem and proficiency.

## 2 Theoretical background

ICALL systems are ITSs, and it is relevant to understand the distinctions between the types of expert models before reviewing the types of ICALL models.

**Expert models in general ITSs.** Intelligent Tutoring Systems require expert models which fall into three main categories, i.e. black-box, glass-box (or rule-based) and cognitive models (Nkambou et al., 2010; Anderson, 2013). Following Alevén (2010), we place rule-based models alongside CBMs as part of the cognitive category. Black box models are said to be inexplicit in their representations as they only provide the final results (Nkambou, 2010, p.18) and show correct input-output behaviour with very little use for their internal computation (Anderson, 2013, p.26).

Cognitive models show different degrees of interpretability, which is useful for instruction delivery. Their decision making processes lend themselves well to giving feedback to learners. In the subcategory of CBMs the requirements that all

solutions should satisfy are set in advance rather than having to map all possible errors and correct solutions. This simplifies the search space by narrowing down the possible solutions and avoiding breaking the domain principles. Conversely, rule-based models rely on an comprehensive set of rules that can be deterministic or probabilistic. The rules mirror the way an expert analyses a problem by taking into account positive or negative observations.

A good expert model seems to revolve around several principles. Not only does it have to produce correct results, but it also needs to have high cognitive fidelity, i.e., the compliance of its decision making features with those that are used by learners. In addition, the expert model must filter out the feature space “according to the same restrictions as a human does” (Anderson, 2013).

**Expert models in ICALL.** The aforementioned distinctions can be used to understand the different types of expert models that exist in ICALL systems. Are they black boxes or cognitive models? Rule-based or CBM? Depending on the tasks and the adopted NLP approaches, they may fall into one of the categories, showing or not their cognitive inclination.

As far as we know, most second language models employ supervised learning methods which rely on very different types of features. Neural approaches with text embeddings and transformer models provide very accurate results in different tasks such as Grammar Error Detection (GED) (Bryant and Briscoe, 2018) or Automatic Essay Scoring (AES) (Rama and Vajjala, 2021). However, the rules and features they rely on are difficult to interpret, turning them into black boxes and leading to poor cognitive fidelity.

A number of GED tasks have relied on supervised learning approaches based on error coded datasets including corrected statements as target hypotheses (Settles et al., 2018). These hypotheses may be seen as the CBM principle, i.e. reference points with which learner language is compared (Bryant et al., 2017). By way of edit-distance metrics, the models can be used to provide error identification in context. However, they cannot explain the reasons for the errors. Their decision making process does not rely on information that is cognitively meaningful for learners.

Other supervised-learning models are based on probabilistic rules relying on explicit linguistic

features. In proficiency prediction tasks, a number of experiments were conducted with models relying on morphosyntactic and lexical features (Tack et al., 2016; Pilán and Volodina, 2018; Yannakoudakis et al., 2018). These linguistic features make up intelligible rules that have a degree of cognitive fidelity. However, in spite of their linguistic characterisation, some are not very actionable by teachers. This is due to the complexity of their design in terms of variables (Gaillat, 2022). For instance, the Automated Readability Index (ARI; (Smith and Senter, 1967)), a measure of difficulty in reading, is composed of two variables (Average Sentence Length (ASL), the Average Word Length (AWL)) whose combination in a formula<sup>1</sup> is hard to interpret. Because they are not designed to provide any other feedback than the result, these models do not have high cognitive fidelity.

Some advanced Automated Writing Evaluation (AWE) systems show greater cognitive fidelity as they try to match their feedback with interpretable linguistic information. Based on linguistic features used in supervised learning methods, the systems can contextualise the errors with grammatical justifications (Attali and Burstein, 2006; Yannakoudakis et al., 2018). Some Automatic Essay Scoring systems, which rely on semantic and discourse complexity metrics, feed from their expert models’ features to elaborate feedback messages on cohesion for learners (Dascalu et al., 2013). In these cases, the connection between the models’ rules and the wording of the feedback messages shows a focus for high cognitive fidelity. While elaborating the messages, these systems rely on expert models that filter out irrelevant knowledge that could impair the cognitive reception by learners. One important aspect is that Dascalu et al. (2013) add specificity as an extra dimension. They use two specific models for two specific tasks, i.e. a view of cohesive links in discourse and a view of stance variation in discourse.

Our proposal follows the same principle applied to the grammatical rather than the cohesive dimension. The objective is to design expert models that capture the hesitations that learner may have on specific syntactic paradigms. For instance, learners may hesitate between different determiners, or they may have confusions in the use of demonstrative pronouns. We intend to design

<sup>1</sup>ARI = 0.5ASL + 4.71AWL - 21.34

several expert models for microsystems specifically linked to linguistic functions. Their goal is to provide fine-grained knowledge of the variations between forms of the same function. Microsystems reflect the learners' hesitations that are part of the competition model in which learners constantly resolve conflicts while choosing forms (MacWhinney et al., 1984). These hesitations create microsystem instability as learners unexpectedly group forms that are not necessarily mapped to the same functional paradigm (Py, 1980). Due to this instability in the mappings, the microsystems are transitional in nature (Gentilhomme, 1980). They include erroneous mappings which later are removed, leading the learner to better proficiency. Following Gaillat et al. (2021), we focus on the referential proform microsystem made up of THIS, THAT and IT. The purpose is to compute how each of these forms, mapped to the same referential function, is likely to occur in relation to its two other competitors.

### 3 Data pre-processing and proform extraction

#### 3.1 Data

The proform microsystem measures are computed with data extracted from the the EF Cambridge Open Language Database (EFCAMDAT) corpus (Geertzen et al., 2013). This corpus results from the collaboration between the Department of Theoretical and Applied Linguistics at the University of Cambridge and Education First (EF). The data was collected on EF EnglishTown, an online school. Our data set is made up of 1,180,507 texts written by students in 191 countries around the world. The data was annotated in terms of 16 proficiency levels which were converted into the six CEFR levels as described in the corpus manual<sup>2</sup>. Table 1 shows the distribution of the average number of words per text and per level.

The data was pre-processed with the methods detailed in Section 3.2. Then, the Grew pattern extraction explained in Section 3.3 was applied, and only the samples where an instance of the microsystem was found are kept. This results in a table that contains 881,627 samples, i.e., as many lines as there are occurrences of proforms IT, THIS and THAT in the EFCAMDAT learner writings. This table also contains 726 columns

<sup>2</sup>Available at [https://corpus.mml.cam.ac.uk/faq/EFCamDat-Intro\\_release2.pdf](https://corpus.mml.cam.ac.uk/faq/EFCamDat-Intro_release2.pdf) (last access 25/03/2023)

CEFR	Writings	Mean of tokens	SD
A1	626,005	39.32	21.46
A2	308,014	68.82	24.42
B1	168,473	98.88	30.23
B2	61,366	137.27	43.67
C1	14,709	171.13	49.03
C2	1,940	176.98	71.95

Table 1: Descriptive statistics of EFCAMDAT writings across CEFR levels

corresponding to the linguistic features about the environment of the microsystem.

#### 3.2 Pre-processing

Prior to microsystem extraction, the data are annotated according to the Universal Dependencies (de Marneffe et al., 2021) framework. The annotations notably include Universal Dependency tagged part-of-speech, lemmas of tokens, and morphological features such as case, number, gender, etc. Linguistic annotations were obtained with the UDPipe pipeline (Straka et al., 2016) using the English model trained on the GUM corpus<sup>3</sup> (Zeldes, 2017). This model shows reliability for POS and dependency annotation on L1 and L2 (Kyle et al., 2022).

#### 3.3 Proform extraction with Grew pattern queries

Grew (Amblard et al., 2022) is a graph rewriting tool that manipulates linguistic representations and is aimed at natural language processing applications. It is used to extract the elements of a microsystem from a sentence, given its linguistic annotations.

Grew creates an annotated graph from a CoNLL-U annotated sentence, with the words and their linguistic annotations (lemma, xpos, upos...) as nodes, and the dependency relations between the words as edges. Using a set of patterns, it is then possible to isolate only the words in the graph that follow these patterns. We create patterns corresponding to proform usage of IT, THIS and THAT. Example (1a) shows the THIS pattern. The heuristic searches for all tokens which are DEPENDENT on a GOVERNOR predicate by a dependency relation of the following types: nominal subject (nsubj) in a passive voice structure (:pass), oblique (obl), nominal modifier, object, conjunct, or root

<sup>3</sup>english-gum-ud-2.5-191206

of sentence (see de Marneffe et al., 2021, p.266-267).

(1a)

```
THIS_PRF::DEP[wordform="this"  
|"these"|"This"|"These"];  
GOV-[nsubj|obl|nsubj:pass|nmod  
|obj|nsubj:outer|conj|root]  
-> DEP.
```

As a results, proforms, such as in examples (2a) and (2b), can be extracted.

- 2a) This song may be a joke now , between musicians , but at the time *it* came , *this* rocked .
- 2b) *That* is how I found the class of Sciences of Education in Paris 2 . I went to the global opening and when I was listening to the presentation of the classes , I was sure *this* was what I wanted to study for my future .

Once the patterns have been extracted, information about the linguistic environment of the target microsystem is collected, including morphological, syntactic and part-of-speech information available for the words in a five-word window around the target proforms. The same type of information is also collected for the dependency governor of the target word, as well as the distance along the dependency tree from the target word to the root of the sentence.

**Evaluation of the extractions** To check the soundness of using Grew patterns to extract microsystems, we conducted the following evaluation. All the sentences that contain an occurrence of the words IT, THIS or THAT, whether they are proforms or not, are selected from the pre-processed data. For each of these words, 100 samples are selected randomly among those that contain the forms, or the maximum amount available if it is less than 100. Additionally, some samples, not containing any of the forms, are also selected. This results in 358 samples used only for evaluation of Grew patterns. This sampling strategy is different from the modelling-evaluation strategy applied in Section 4.3 because, here, it is essential to capture forms of any function. On the contrary, the modelling strategy solely requires proform samples of the forms.

The gold standard is set by an expert who annotates whether the identified form is indeed a IT,

THIS or THAT proform, or none of those. The samples are also independently run through Grew. The tool outputs the patterns for the identified forms which are then compared to the expert annotations. A notable feature of this data is the unbalance of the forms, with the THIS proform making out only 2% of the annotated samples, and 60% of the samples displaying no use of the microsystem. Note that a subsequent development of this study will include three annotators with measures of inter-rater agreements.

The weighted F1-score of Grew extractions reaches 0.82, with a weighted precision of 0.90 and a weighted recall of 0.80. This shows that using Grew patterns as a tool to identify the microsystems is viable, and does not select many forms that are not indeed part of the microsystem. A word of caution is to be given about the results for each individual form: while the IT proform occurrences are almost always perfectly identified, and most THAT forms found by Grew are indeed correct, many relevant THAT proforms are not identified in the text, as shown in Figure 1. This phenomenon can be explained by the sample selection strategy : sentences that contained the string of characters THAT were selected in this data set. However, the word THAT covers a wide range of other functions than proform, namely, subordinator, relativizer, adverbial, demonstrative determiner. IT, on the other hand, is more often used in its referential function in spite of its possible other functions, i.e., impersonal use, extrapositional use, cleft use and expressing weather/time/distance (Huddleston and Pullum, 2002, p. 960, Biber et al., 1999, p. 332). The samples containing THAT are therefore less likely to contain a large proportion of proform use of THAT, contrary to the samples containing IT. On inspecting extraction errors of THAT, it also appears that the proform use is confused with the relativizer use of the form. To address this problem, the Grew extraction query of THAT proform should be revised with a finer-grained filtering strategy. Still, with a grain of salt concerning the extraction of THAT, these results show that Grew is a relevant tool for the extraction of the elements of the microsystem.

Moreover, this first evaluation also provides some insight on the rarity of proform uses of THIS, THAT and IT, highlighting variability in the frequencies of use. IT is more often used in its

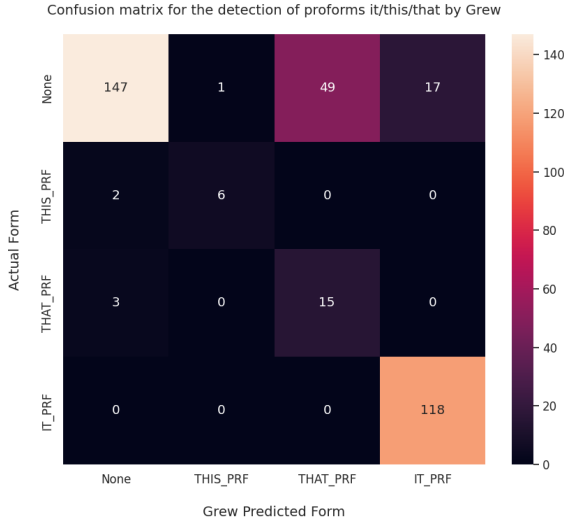


Figure 1: Confusion matrix for the evaluation of Grew pattern extraction

proform function than THAT, for instance. This draws the attention on the need to address this imbalance in crafting the statistical models, as such models are often biased by unbalanced data, and to analyze our results in the light of these uses of the proforms.

## 4 Design of the proform microsystem measures

### 4.1 Conceptual design

The conceptual idea of a microsystem is that each form is used relative to its competitor forms because they are mapped to the same referential function. For instance, one possible assumption about the proform microsystem is that the use of a THAT is detrimental to the use of IT and THIS. In order to identify the best operationalisation of the microsystem concept, we identified three types of measures capturing the forms' relative variations. The measures are based either on proportions or probabilities of occurrence.

Regarding proportions, we tally the counts of each IT, THIS and THAT for each writing, then create the percentage for each MS as in:

$$MS(x_{ij}) = f_{ij} / \sum_{k=1}^n f_{kj}$$

where  $x$  = the microsystem,  $i$  = the  $i$ th component of the microsystem,  $j$  = the  $j$ th text,  $n$  = the total number of forms in microsystem and  $f_{ij}$  = the frequency of a component in text  $j$ .

Regarding probabilities, we apply two types of models. First, a multinomial logistic regression model predicting forms on the basis of linguistic

features extracted from the forms' local contexts.

$$\sigma^{-1}(p(i|C)) = \alpha + \beta_1(c_1) + \dots + \beta_n(c_n) + \epsilon$$

where  $p(i|C)$  is the probability of observing a proform  $i$  knowing the context  $C$  made up of features  $C = \{c_1, \dots, c_n\}$  and  $\sigma^{-1}$  is the logit function.

Second, a neural network predicting the probabilities of a form given the linguistic environment. Given an input sample  $C$  that represents the linguistic environment of a form, the goal is to compute the conditional probability of observing one of the forms of the microsystem, i.e. THIS, THAT or IT.

$$p(i|C) = \sigma(f_2(LR(f_1(C))))$$

where  $p(i|C)$  is the probability of observing a proform  $i$  knowing the context  $C$  as defined above,  $f_k(c) = cA_k + b_k$  are linear layers with trainable parameters  $A_k$  and  $b_k$ ,  $LR(c) = \max(0, c) - 0.01 \times \min(0, c)$  is a Leaky ReLU activation function, and  $\sigma(c)_i = e^{c_i} / \sum_{j=1}^K e^{c_j}$  is the softmax activation function. The input  $C$  consists of the one-hot-encoded categorical variables in the linguistic environment of a form. The LeakyReLU activation function has been preferred over the ReLU function as a way to mediate the issues of vanishing gradient during training, induced by the sparse feature representation of the input due to one-hot-encoding.

### 4.2 Technical implementation

The relative proportions are based on the raw frequencies of the proforms in each text and are computed on all the texts.

In the case of the logistic regression measuring approach, the model relies on the following features: POS, Universal Dependency information regarding heads, POS of tokens found in a [-5;+5] position interval and dependency distance between a form and its head. As not all variables of the data set were assigned values (especially morphological features which are dependent on word types) variables with more than 10% missing values are dropped.

In the case of the Neural measuring approach, all available linguistic annotation is collected as features at first: POS, morphological features and Universal Dependency information of tokens found in the [-5;+5] position window and dependency distance between a form and its head. Then, only features where more than 60% of the samples are not null were kept. This is done as an

	A1	A2	B1	B2	C1	C2	Total
IT	18360	20291	18406	11482	2476	371	71386
THAT	9268	16320	25009	14950	5009	830	71386
THIS	19372	16821	20673	10565	3518	437	71386
Total	47000	53432	64088	36997	11003	1638	214158

Table 2: Number of samples for each proform at each CEFR level in the balanced training data set

attempt to reduce unnecessary dilution of the information through one-hot-encoded variables that would be mostly null. The network is trained over 50 epochs, using the Adam optimiser with a 0.0005 learning rate.

### 4.3 Evaluation

To evaluate the accuracy of the measures given by the proform predictive approaches, we split the data into 80% training and 20% testing, for a total of 705,302 training samples and 176,325 test samples. The training data set is then balanced with regard to the number of THIS, THAT or IT forms, in order to avoid model imbalance. We take random samples of IT and THAT equal to the number of THIS (i.e., 71,386 occurrences, the lowest number among the three proforms), resulting in 214,158 training samples. Details on the composition of the training data set are provided in Table 2.

We evaluate the three measure construction methods, that is, proportions, logistic model probabilities and neural network probabilities, in two steps. Firstly, we examine the predictive capacity of the systems used to create the measures : if these models cannot properly classify the proform given a certain context, then they are not likely to create good measures for the microsystem. We therefore perform multinomial logistic regression and train the neural network approach on the training data and predict labels in the testing data, using linguistic features listed in Section 4.2.

In a second phase, to evaluate whether the measures correlate with proficiency, we perform modelling with ordinal logistic regression as a descriptive model. Taking as descriptors the probabilities of using THIS, THAT and IT output by the previous model, we investigate whether there is an association between these measures and the odds of increasing CEFR level.

## 5 Results and discussion

### 5.1 Measure creation

We separately inspect the three approaches used to create the measures. The first proportion-based

approach can only provide an insight in the tendency of the learners to use the different forms of the microsystem, as it is a count-based method and not a statistical model. The other two approaches can be evaluated with the usual accuracy, prediction and recall scores, presented in Table 3.

Beginning with measures based on proportions, Figure 2 depicts the distribution of IT, THIS, THAT relative proportions across CEFR levels. It shows a reduction in the percentage use of IT as CEFR level increases. The reverse is seen in the percentage of THAT use, with an increase at higher CEFR levels. A Kruskal-Wallis rank sum test (or “one way ANOVA on ranks”) is used to quantify the differences between MS proportions at different levels. The p-value smaller than 0.05 for all three proforms ( $p < 0.01$ ) indicates significant differences between the use of proforms at different CEFR levels.

Moving on to the second approach, the multinomial logistic regression yields a 0.77 accuracy overall (95% CI: (0.76, 0.77),  $p > .001$ ). The detailed results in Table 3 show reasonable accuracy statistics for IT microsystems but low recall and precision for THIS and THAT proforms. The difficulty in picking THAT with local context features might come from the higher complexity of the THAT contexts of occurrence due to the form’s functional versatility, i.e. subordinator, relativizer, adverbial, demonstrative determiner and proform.

Thirdly, the model based on the neural network yields a 0.74 accuracy (95% CI: (0.73, 0.74),  $p > 0.001$ ). This model shows the same issues as the multinomial logistic regression model regarding the prediction of THIS, although it performs slightly better. However, the performance increases for THAT, with the recall more than doubling. This could be explained by the capacity of the neural network to create a high-dimensional latent feature space, where the different functions of THAT crystallise over different dimensions, disambiguating the use of THAT as proform as a result.

The best performances overall are therefore reached by the neural network approach, although the multinomial regression method offers better performance for IT alone, and the proportions are proved to show statistically significant differences between CEFR levels. In order to leverage the advantages of these three approaches, a possible avenue for future work is to explore a combination of



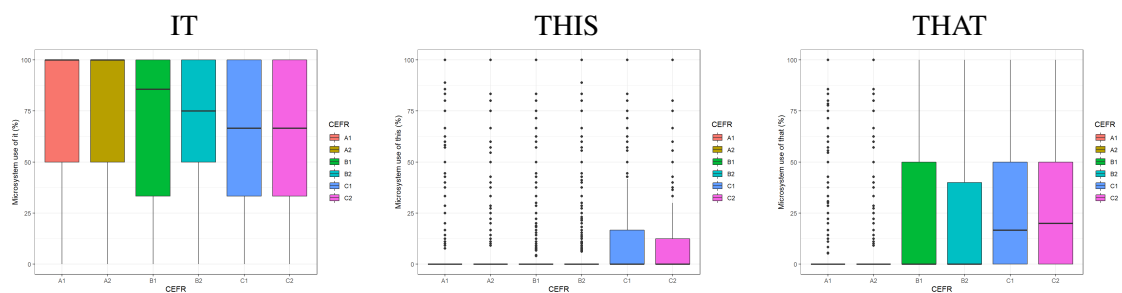


Figure 2: Distribution of relative proportions of IT, THIS and THAT proforms across CEFR levels in the EFCAM-DAT corpus

these models. It could either be a simple concatenation of all measures, leading to a 9 microsystem measures, or a weighted sum of the measures, with weights proportional to the performance of each model on each specific form. Another approach could also lie with bagging several multinomial regressions, and using the average of the output probabilities as our measures.

## 5.2 Association between measures and CEFR level

We now report the results regarding the models computing associations between the measures and the odds of increasing CEFR levels, using an ordinal regression analysis. Table 4 shows odds ratios for each of the three types of measure. For all measures, the odds ratios are significant ( $p < 0.001$ ), comforting the fact that our microsystem measures can be used as predictors of CEFR level.

These results suggest that writings with higher predicted probability of IT have a reduced odds of being in a higher CEFR level. On the contrary, those with a higher predicted probability of THAT are more likely to have a higher CEFR level. Both predictive methods (multinomial regression and neural networks) agree on a higher probability of THIS being more likely to have a higher CEFR level, while the proportions method finds that on the contrary, a higher proportion of THIS hints at a lower CEFR level. We believe that in this case, this is due to a limitation of the proportion-based measure, that simply counts the percentage of occurrence of each form regardless of linguistic context. The proportion of THIS contains many more outliers than the other two forms, as seen in Figure 2. Our explanation is that the disagreement with the other two models is caused by an inaccurate measure of THIS due to this scattered distribution.

## 5.3 Discussion

Our first research question revolved around the likelihood of microsystem forms in L2 writings. The three measurement methods we propose capture the choices of a given form with regard to the other possible forms. Regarding model performance, IT is always well predicted while the detection of THIS and THAT could be improved, leading to more accurate probabilities and in turn better microsystem descriptors. To this end, more significant features defining the local context of occurrence of the proforms could be assessed. For instance, adding referential information regarding the degree of givenness of a proform could possibly improve the models. Another way to improve the contextualization of the proforms could be the use of state-of-the-art Natural Language Processing approaches, such as Long-Short Term Memory networks (Hochreiter and Schmidhuber, 1997), or more recently, BERT models (Devlin et al., 2019). Both these methods could be used in the same fashion as we did in the present work, that is, trained to predict a masked form, with the additional benefits of feeding the entire text "as is" to the model and not needing to hand-craft context features. It must be noted that our use of a neural model differs from black-box models that rely on the direct ingestion of texts to predict errors for instance. Our neural model relies on proforms rather than full texts, hence giving specific grammatico-functional probabilities that can be used in subsequent higher-level prediction tasks.

The second research question was to analyse the degree of association between the measures and the CEFR levels given to the texts. Our results indicate that an expert proform MS model can be trained on the basis of likelihood of occurrence, with a slight disagreement between proportion-based and probability-based measures. In both cases, an expert model could use these two mea-

	Multinomial log regression			Neural network		
	IT	THIS	THAT	IT	THIS	THAT
Balanced accuracy	0.71	0.69	0.64	0.74	0.70	0.72
Precision	0.93	0.20	0.36	0.93	0.31	0.66
Recall	0.81	0.54	0.31	0.74	0.70	0.72

Table 3: Performance statistics for the predictive approaches to measuring the proform microsystem

		Odds ratio	95% CI	p_value
Proportions	IT	0.995	0.995, 0.995	<0.001
	THIS	0.997	0.997, 0.997	<0.001
	THAT	1.010	1.010, 1.010	<0.001
Multinom log regression	IT	0.992	0.991, 0.993	<0.001
	THIS	1.006	1.005, 1.007	<0.001
	THAT	1.012	1.011, 1.014	<0.001
Neural network	IT	0.47	0.42, 0.51	<0.001
	THIS	1.17	1.04, 1.32	<0.001
	THAT	2.27	2.04, 2.53	<0.001

Table 4: Ordinal logistic regression of CEFR by proportion of IT,THIS and THAT

asures as predictors of CEFR levels in new incoming learner writings.

The MS model also supports qualitative feedback with regards to specificity and cognitive fidelity. Firstly, the probability-based models offer knowledge of proform use at word level, allowing specific identification in context, hence specific feedback. A high level of feedback specificity improves understanding from the learner (Shute, 2008). Secondly, because of the grammatico-functional nature of the MS concept, the MS model’s measures can be used to explain reasons of a problem. For instance, a proficiency-predicting model relying on MS proform features could point out the demonstrative pronouns in a learner’s text in a similar fashion to what Dascalu et al. (Dascalu et al., 2013) do by identifying cohesion gaps. This level of explainability gives a high degree of cognitive fidelity. In this respect, the neural-model increases interpretability as it provides a broader variation of odds ratios, indicating clearer proficiency gaps and making the effects of each form clearer to disambiguate.

## 6 Conclusion

In this paper, we have reported on the design of new grammatico-functional metrics which are to be used in the expert module of an ICALL system. The metrics rely on paradigmatic syntactic relations between words of specific functions. We have illustrated the approach with the design of the IT, THIS, THAT proform microsystem. The measures rely on the relative proportions of the forms and their likelihood of occurrence. They show sig-

nificance in terms of association with CEFR levels, paving the way for their introduction in a specific proform microsystem expert model.

## 7 Acknowledgments

This project is funded by the French National Research Agency. ANR-22-CE38-0015-01



## References

- Vincent Alevan. 2010. [Rule-Based Cognitive Modeling for Intelligent Tutoring Systems](#). In Roger Nkambou, Jacqueline Bourdeau, and Riichiro Mizoguchi, editors, *Advances in Intelligent Tutoring Systems*, number 308 in *Studies in Computational Intelligence*. Springer Berlin Heidelberg.
- Maxime Amblard, Bruno Guillaume, Siyana Pavlova, and Guy Perrier. 2022. [Graph querying for semantic annotations](#). In *Proceedings of the 18th Joint ACL - ISO Workshop on Interoperable Semantic Annotation within LREC2022*, pages 95–101. European Language Resources Association.
- John R. Anderson. 2013. The Expert Module. In *Foundations of Intelligent Tutoring Systems*, pages 21–54. Psychology Press.
- Yigal Attali and Jill Burstein. 2006. [Automated Essay Scoring With e-rater® V.2](#). *The Journal of Technology, Learning and Assessment*, 4(3):3–29.
- Douglas Biber, Stig Johanson, Geoffrey Leech, Susan Conrad, and Edward Finegan. 1999. *Longman Grammar of Spoken and Written English*. Longman, Harlow.
- Christopher Bryant and Ted Briscoe. 2018. [Language Model Based Grammatical Error Correction without Annotated Training Data](#). In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 247–253, New Orleans, Louisiana. Association for Computational Linguistics.
- Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. [Automatic Annotation and Evaluation of Error Types for Grammatical Error Correction](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long*

- Papers*), pages 793–805, Vancouver, Canada. Association for Computational Linguistics.
- Bram Bulté and Alex Housen. 2012. *Defining and Operationalising L2 Complexity*. John Benjamins Publishing Company.
- Mihai Dascalu, Philippe Dessus, Stefan Trausan-Matu, Maryse Bianco, Aurélie Nardy, Mihai Dascălu, and Ștefan Trăușan-Matu. 2013. *ReaderBench, an Environment for Analyzing Text Complexity and Reading Strategies*. In *AIED 13 - 16th International Conference on Artificial Intelligence in Education*, volume 7926 of *Lecture Notes in Computer Science (LNCS)*, pages 379–388, Memphis, TN, United States. Springer.
- 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.
- Thomas Gaillat. 2022. *Investigating the scopes of textual metrics for learner level discrimination and learner analytics*. In Agnieszka Leńko-Szymańska and Sandra Götz-Lehmann, editors, *Complexity, Accuracy and Fluency in Learner Corpus Research*, number 104 in *Studies in Corpus Linguistics*, pages 21–50. John Benjamins.
- Thomas Gaillat, Andrew Simpkin, Nicolas Ballier, Bernardo Stearns, Annanda Sousa, Manon Bouyé, and Manel Zarrouk. 2021. *Predicting CEFR levels in learners of English: The use of microsystem criterial features in a machine learning approach*. *RECALL*, 34(2).
- Jeroen Geertzen, Theodora Alexopoulou, Anna Korhonen, et al. 2013. Automatic linguistic annotation of large scale l2 databases: The ef-cambridge open language database (efcamdat). In *Proceedings of the 31st Second Language Research Forum, Somerville, MA: Cascadilla Proceedings Project*, pages 240–254.
- Yves Gentilhomme. 1980. Microsystèmes et acquisition des langues. *Encrages*, pages 79–84.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Rodney Huddleston and Geoffrey K. Pullum. 2002. *The Cambridge Grammar of The English Language*. Cambridge University Press, Beccles, Suffolk.
- Kristopher Kyle, Masaki Eguchi, Aaron Miller, and Theodore Sither. 2022. *A Dependency Treebank of Spoken Second Language English*. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 39–45, Seattle, Washington. Association for Computational Linguistics.
- Anke Lüdeling and Hagen Hirschmann. 2015. Error Annotation Systems. In Sylviane Granger, Gaëtanelle Gilquin, and Fanny Meunier, editors, *The Cambridge Handbook of Learner Corpus Research*, pages 135–158. Cambridge University Press, Cambridge.
- Brian MacWhinney, Elizabeth Bates, and Reinhold Kliegl. 1984. *Cue validity and sentence interpretation in English, German, and Italian*. *Journal of Verbal Learning and Verbal Behavior*, 23(2):127–150.
- Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. *Universal Dependencies*. *Computational Linguistics*, (2):255–308.
- Roger Nkambou. 2010. *Modeling the Domain: An Introduction to the Expert Module*. In Roger Nkambou, Jacqueline Bourdeau, and Riichiro Mizoguchi, editors, *Advances in Intelligent Tutoring Systems*, number 308 in *Studies in Computational Intelligence*, pages 15–32. Springer Berlin Heidelberg.
- Roger Nkambou, Jacqueline Bourdeau, and Riichiro Mizoguchi, editors. 2010. *Advances in Intelligent Tutoring Systems*. Number 308 in *Studies in Computational Intelligence*. Springer Berlin Heidelberg.
- Anne O’Keeffe and Geraldine Mark. 2017. *The English Grammar Profile of learner competence: Methodology and key findings*. *International Journal of Corpus Linguistics*, 22(4):457–489. Publisher: John Benjamins.
- Ildikó Pilán and Elena Volodina. 2018. *Investigating the importance of linguistic complexity features across different datasets related to language learning*. In *Proceedings of the Workshop on Linguistic Complexity and Natural Language Processing*, pages 49–58, Santa Fe, New-Mexico. Association for Computational Linguistics.
- Bernard Py. 1980. Quelques réflexions sur la notion d’interlangue. *Revue Tranel (Travaux neuchâtelois de linguistique)*, 1:31–54.
- Taraka Rama and Sowmya Vajjala. 2021. *Are pre-trained text representations useful for multilingual and multi-dimensional language proficiency modeling?* ArXiv:2102.12971 [cs].
- Burr Settles, Chris Brust, Erin Gustafson, Masato Hagiwara, and Nitin Madnani. 2018. *Second Language Acquisition Modeling*. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 56–65, New Orleans, Louisiana. Association for Computational Linguistics.
- Valerie J. Shute. 2008. *Focus on formative feedback*. *Review of Educational Research*, 78(1):153–189.

- Edgar A. Smith and R. J. Senter. 1967. Automated readability index. Technical Report AMRL-TR-66-220, Aerospace Medical Division, Wright-Paterson AFB, Ohio.
- Milan Straka, Jan Hajič, and Jana Straková. 2016. UD-Pipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, pages 4290–4297. European Language Resources Association.
- Anaïs Tack, Thomas François, Anne-Laure Ligozat, and Cédric Fairon. 2016. Evaluating Lexical Simplification and Vocabulary Knowledge for Learners of French: Possibilities of Using the FLELex Resource. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 230–236, Portorož, Slovenia. European Language Resources Association (ELRA).
- Helen Yannakoudakis, Øistein Andersen, Ardeshir Geranpayeh, Ted Briscoe, and Diane Nicholls. 2018. Developing an automated writing placement system for ESL learners.
- Amir Zeldes. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

# Japanese Lexical Complexity for Non-Native Readers: A New Dataset

Yusuke Ide<sup>1</sup> Masato Mita<sup>2</sup> Adam Nohejl<sup>1</sup> Hiroki Ouchi<sup>1,3</sup> Taro Watanabe<sup>1</sup>

<sup>1</sup>Nara Institute of Science and Technology <sup>2</sup>CyberAgent Inc. <sup>3</sup>RIKEN  
{ide.yusuke.ja6, nohejl.adam.mt3, hiroki.ouchi, taro}@is.naist.jp,  
mita\_masato@cyberagent.co.jp

## Abstract

Lexical complexity prediction (LCP) is the task of predicting the complexity of words in a text on a continuous scale. It plays a vital role in simplifying or annotating complex words to assist readers. To study lexical complexity in Japanese, we construct the first Japanese LCP dataset. Our dataset provides separate complexity scores for Chinese/Korean annotators and others to address the readers' L1-specific needs. In the baseline experiment, we demonstrate the effectiveness of a BERT-based system for Japanese LCP.

## 1 Introduction

Reading comprehension requires a certain level of vocabulary knowledge. The results reported by Hu and Nation (2000) suggest that most English learners need to understand 98% of tokens in a text to comprehend it. A follow-up study by Komori et al. (2004) estimates the percentage to be 96% for Japanese learners to comprehend text. Acquiring vocabulary to reach such levels, in turn, is a lengthy and challenging task for learners. This opens up opportunities for assistive applications, such as simplification or annotation of complex words. The first step necessary for such applications is to predict the complexity of the words. The task of **lexical complexity prediction (LCP)** is defined as predicting how difficult to comprehend words or phrases in a text are on a continuous scale (Shardlow et al., 2020). This differentiates LCP from complex word identification (CWI), i.e., binary classification of complex words (Yimam et al., 2018). As complexity is naturally perceived as continuous, a continuous scale used in LCP allows to represent it without loss of information.

The LCP research so far has been limited to English, for which two LCP datasets have been constructed (Shardlow et al., 2020, 2022), and no such dataset has been created for Japanese. Meanwhile, there are a number of features specific to the

Japanese language that could affect lexical complexity, and their effects have yet to be studied. For example, the Chinese characters, which are used extensively in Japanese, lower text readability (Tateisi et al., 1988).

Previous studies on Japanese lexical complexity used pedagogical word lists to estimate complexity level. Nishihara and Kajiwara (2020) modeled lexical complexity of words based on the Japanese Educational Vocabulary List (Sunakawa et al., 2012). The word list assigns a degree of difficulty to each item, based on the subjective judgment of Japanese language teachers, not learners themselves, and does not consider the learners' L1 background.

In light of this, we present JaLeCoN<sup>1</sup>, Dataset of **Japanese Lexical Complexity for Non-Native Readers**. Our dataset has the following key features:

- (1) Complexity scores for single words as well as multi-word expressions (MWEs);
- (2) Separate complexity scores from Chinese/Korean annotators and others, addressing the considerable advantage of the former in Japanese reading comprehension.

Our analysis reveals that the non-Chinese/Korean annotators perceive words of Chinese origin or containing Chinese characters as especially complex. In the baseline experiment, we investigate the effectiveness of a BERT-based system in the Japanese LCP task, and how it varies according to the word complexity and L1 background.

## 2 Task Setting

Since Japanese has no explicit word boundaries, word segmentation is the first prerequisite for LCP. We use short unit words (SUWs) as the basic word unit, combining them into longer word units in the case of multi-word expressions (MWEs):

<sup>1</sup>JaLeCoN is available at <https://github.com/naist-nlp/jalecon>.

<b>SUWs</b>	右肩	上がり	に	増え	て	いる
	right.shoulder	rise	ADV	increase	GER	be-PRS
<b>Words</b>	<div style="border: 1px solid orange; padding: 2px; display: inline-block;">       右肩上がり MWE        steady.rise     </div>		<div style="border: 1px solid blue; padding: 2px; display: inline-block;">       に SUW        ADV     </div>	<div style="border: 1px solid blue; padding: 2px; display: inline-block;">       増え SUW        increase     </div>	<div style="border: 1px solid orange; padding: 2px; display: inline-block;">       ている MWE        PRG-PRS     </div>	
	“is steadily increasing”					

Figure 1: Example of text segmented as SUWs and as words (either SUW or MWE). Semantically opaque sequences are chunked into MWEs. Abbreviations in glosses: ADVerbializer, GERund, PRoSent, PRoGressive.

**SUW:** SUWs consist of one or two smallest lexical units (Ogura et al., 2011), and are commonly used for segmentation of Japanese.

**MWE:** We understand MWEs as multi-SUW expressions that are fixed or semantically opaque (see Appendix C) and consequently may have higher complexity than their components. We identify MWEs either using long unit word (LUW)<sup>2</sup> segmentation, or manually (see Section 3).

Consequently, a **word**, can be either an SUW or an MWE (see Figure 1 for examples).

A **complexity score** represents perceived complexity based on the annotators’ judgment on a scale from 0 (least complex) to 1 (most complex). We exclude proper nouns from our target because their complexity is influenced by factors unrelated to reading proficiency or vocabulary knowledge.<sup>3</sup>

We annotate the words in an **in-context dense** setting. In-context here means including both intra-sentence and extra-sentence context of each word. Context is important for lexical complexity for two reasons (Gooding and Kochmar, 2019; Shardlow et al., 2021): (1) As polysemous words can have different complexity levels for each sense, context is necessary to differentiate between possible meanings of these words. (2) Presenting a word without context could increase its complexity. In particular, the recognition of abstract words relies on context (Schwanenflugel et al., 1988). **Dense** means annotating each word of the text with a complexity label, instead of annotating one specific word in each sentence (Shardlow et al., 2022). We adopt the dense setting to avoid any bias that could arise from targeting specific words.

### 3 Construction of JaLeCoN

In order to include both written and spoken language and a variety of vocabulary, we sourced texts

<sup>2</sup>The LUW is defined as a syntactic word by Omura et al. (2021).

<sup>3</sup>Sequences containing segmentation errors are also excluded (see Appendix D).

from two different genres:

**News** comes from the Japanese-English data of the WMT22 General Machine Translation Task (Kocmi et al., 2022). It contains a variety of news texts written for the general Japanese reader.

**Government** is composed of press conference transcripts from Japanese ministries or agencies.<sup>4</sup>

The whole dataset is composed of sequences of sentences constituting either the beginning of an article (News) or a question-answer pair (Government). We restricted the length of the sequences to at least 6 and at most 11 sentences to obtain similar amounts of text, and presented each sequence as a whole for annotation.

#### 3.1 Word Segmentation

We used Comainu 0.80<sup>5</sup> (Kozawa et al., 2014) to perform two-level segmentation. The low-level SUW segmentation was done using MeCab (Kudo et al., 2004), a Japanese morphological analyzer, and the UniDic 2.3.0 (Den et al., 2007) dictionary. At the second level, Comainu chunked the SUWs into LUWs. Based on the two segmentations, we segmented the text into words as follows:

(1) If an LUW is a noun, we use the constituting SUWs as words. Transparent noun compounds are ubiquitous in Japanese (e.g., 次期 | 気象 | 衛星<sup>6</sup> “next meteorological satellite”), and we do not consider them MWEs.

(2) If an LUW is not a noun, we use the LUW as a word. Such an LUW may be a single SUW, or a sequence of SUWs, which we consider an MWE. Such MWEs most importantly include functional words, such as compound particles (e.g., に | 比べ | て “compared to”) and auxiliary verbs (e.g., な | けれ | ば | なら | ない “have to”).

We also identified other MWEs manually, as explained in Section 3.3.

<sup>4</sup>The transcripts were retrieved from the websites of five organizations: JMA, JTA, MOJ, MOFA, and MLHW.

<sup>5</sup><https://github.com/skozawa/Comainu>

<sup>6</sup>The vertical bars denote boundaries between SUWs.

Genre	Sentences	Words	MWE Ratio	CK		Non-CK	
				All Words	MWEs	All Words	MWEs
News	400	10,256	7.9%	.009	.020	.024	.072
Government	200	7,964	14.4%	.005	.009	.028	.047

Table 1: Statistics of JaLeCoN. The CK and Non-CK columns show the mean complexity scores by L1 group.

### 3.2 Complexity Annotation

To capture the lexical complexity for a non-native Japanese reader with intermediate or advanced reading ability, we recruited 15 annotators per sentence with Japanese reading proficiency ranging from CEFR (Common European Framework of Reference for Languages) level B1 to C2. We required at least intermediate proficiency, as it has been shown that complexity judgments made by intermediate or advanced learners can be used to adequately predict the needs of beginners but not vice versa (Gooding et al., 2021). The proficiency levels were self-reported (see Appendix A for details). We used the annotations made by 14 of them, after removing one outlier, whose annotations had over 70% higher mean than those of any other annotator, clearly not corresponding to the reported reading proficiency.

Approximately half of the annotators we recruited have a Chinese/Korean L1 background (CK).<sup>7</sup> CK learners have a considerable advantage in comprehension of words of Chinese origin, which also form a large part of Chinese and Korean vocabulary (Koda, 1989).

The annotators were asked to assign one of the following labels to each span if they find it complex: 3 (Very Difficult), 2 (Difficult), or 1 (Not Easy); otherwise the annotators were to leave the span unlabeled and we interpreted it as 0 (Easy).<sup>8</sup> Annotators could label a span of any length if it was complex as a whole, but were asked to create as short a span as possible. To calculate the average, the labels were converted to numerical values as follows: 3  $\rightarrow$  1, 2  $\rightarrow$  0.67, 1  $\rightarrow$  0.33, 0  $\rightarrow$  0. The averaging hinges on the assumption that the labels have an equal distance between them. We always presented the labels together with the values 0 to 3 to reinforce the perception of equal distance.

<sup>7</sup>On average, the CK annotators reported higher Japanese reading proficiency than the non-CK (see Appendix A).

<sup>8</sup>See Appendix B for detailed definitions of each label.

### 3.3 MWE Annotation

In parallel with the complexity annotation, we annotated MWEs not identified by LUW segmentation (see Section 3.1). Given the absence of an MWE detector for Japanese of sufficient quality, the annotation was performed manually by a native Japanese speaker and a non-native speaker with a degree in the Japanese language. The expression categories we consider MWEs are described in Appendix C.

### 3.4 Complexity Scoring

Using annotations from the previous steps, we assigned complexity scores to words according to the following rules:

- (1) If a span contains one or more words, each word receives the complexity value of the span.
- (2) If an MWE (manually annotated according to Section 3.3) overlaps with or contains multiple spans, the MWE receives the maximum of the complexity values of the spans.

Finally, for each word, we calculated the complexity score for each L1 group as the average of the individual values from the annotators in that group.

## 4 Statistics and Analysis

Overall statistics for both genres and L1 groups are shown in Table 1.<sup>9</sup> MWEs have higher mean complexity than single words for both L1 groups and are more frequent in the Government genre. There is a tendency towards perceiving higher complexity in the non-CK group, which corresponds to slightly lower average Japanese proficiency of the non-CK annotators (see Appendix A).

We measured inter-annotator agreement (IAA) using Krippendorff’s  $\alpha$  for interval values (Krippendorff, 1970). The IAA is 0.32 in the CK group, and 0.31 in the non-CK group, while it would be 0.19 if we merged the groups. As lexical complexity is

<sup>9</sup>See Appendix E for the complexity scores and annotation distributions of several words in the non-CK group.

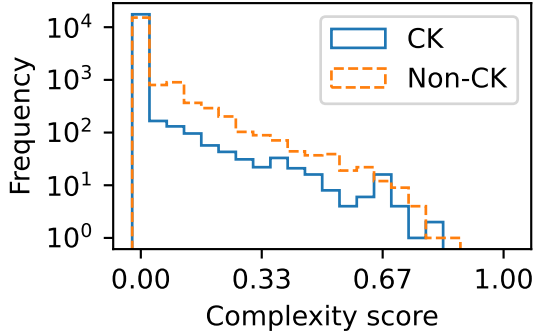


Figure 2: Histogram of complexity scores by L1 group.

	Japanese		Chinese		Other	
	All	CC	All	CC	All	CC
CK	.003	.009	.004	.004	.071	.000
Non-CK	.010	.032	.062	.072	.007	.143
Frequency	52%	10%	26%	22%	4%	0%

Table 2: Mean complexity (by L1 group) and frequency, according to (1) word origin: Japanese (*wago*), Chinese/Sino-Japanese (*kango*), and Other (*gairaigo*, borrowings from languages other than Chinese), and (2) whether the words contain Chinese characters (denoted by CC). The origin was classified using MeCab and Comainu (see Section 3.1), excluding words of mixed or unknown origin.

highly subjective (Gooding et al., 2021), the low agreement does not imply low reliability, but it indicates that perception of complexity is more alike within the L1 groups than across all annotators.

The complexity score distribution in each L1 group is shown in Figure 2. No words achieved a score greater than 0.81 and 0.86 in the CK and non-CK groups, respectively, which reflects that words are rarely labeled as Difficult or Very Difficult by all annotators in a group.

In addition to the aforementioned difference in proficiency, there is also a clear difference in how the two L1 groups perceive complexity of words based on their origin and whether they contain Chinese characters<sup>10</sup>, as analyzed in Table 2. For the CK group, the mean complexity of words of Japanese and Chinese origin was similar. For the non-CK group, however, words of Chinese origin

<sup>10</sup>Japanese vocabulary consists of words of Japanese origin, Chinese (Sino-Japanese) origin, and foreign words from other languages (*gairaigo*). The first two categories can be written using Chinese characters (*kanji*), Japanese syllabary (*kana*), or a combination thereof, while other foreign words are usually written in syllabary only. (See Appendix F for examples.)

were markedly more complex (0.062) than words of Japanese origin (0.010), and both categories of words were more complex when they contained Chinese characters.<sup>11</sup>

## 5 Experiments

The newly created dataset can be used to evaluate performance of LCP for non-native Japanese readers of different L1 backgrounds (CK and non-CK). We developed a baseline system based on a fine-tuned BERT (Devlin et al., 2019) model, and evaluated it using cross-validation. We fine-tuned a Japanese pre-trained BERT model released by Tohoku University, namely the base model for UniDic Lite segmentation<sup>12</sup>.

For each word  $w$  in our dataset and the sentence  $s$  that contains it at token indices  $i$  to  $j - 1$ , we construct an input sequence ( $[\text{CLS}], s_0^{i-1}, \langle \text{Unused1} \rangle, w, \langle \text{Unused2} \rangle, s_j^{j-1}, [\text{SEP}], w, [\text{SEP}]$ ). The target word occurs first delimited by unused tokens ( $\langle \text{Unused}n \rangle$ ) in the sentence context, and then on its own following the first  $[\text{SEP}]$  token.<sup>13</sup> To predict the complexity score, we feed the final hidden representation of the  $[\text{CLS}]$  token into a linear layer with a single output. A similar fine-tuning approach, but without the special tokens, was used for English LCP by Taya et al. (2021), achieving one of the highest  $R^2$  values in the single-word subtask of SemEval-2021 Task 1 (Shardlow et al., 2021).

We fine-tune and evaluate models for CK and non-CK complexity separately. See Appendix G for the hyperparameters and cross-validation scheme.

The results are reported in Table 3. In addition to  $R^2$  (coefficient of determination)<sup>14</sup>, we report mean average error (MAE) by complexity score tiers to draw the full picture of the models’ performance at different complexity levels. The score ranges of the

<sup>11</sup>The opposite tendency for *gairaigo* (foreign words mostly from English) to be perceived as more complex in the CK group coincides with lower English proficiency among annotators in this group (see Appendix A), and therefore should not be explained by their L1 background.

<sup>12</sup>Available from <https://huggingface.co/cl-tohoku/bert-base-japanese-v2>.

<sup>13</sup>Due to a different segmentation (version of UniDic) used by Tohoku BERT and our dataset, we have to enforce segmentation at the word’s boundaries using spaces.

<sup>14</sup>Compared to correlation coefficients,  $R^2$  is more appropriate for LCP, since it also captures deviations in mean and variance. Compared to MAE or MSE, it is easier to interpret, as  $R^2 = 0$  corresponds to the mean regressor, while  $R^2 = 1$  corresponds to a perfect model.



MAE by Gold Complexity Score Tier					
	<u>Zero</u>	<u>Easy &gt; 0</u>	<u>Not Easy</u>	<u>(Very) Difficult</u>	$R^2$
CK	0.0034	0.0676	0.1913	0.2954	0.4351
Non-CK	0.0066	0.0510	0.1169	0.2932	0.6142

Table 3: Results of the fine-tuned BERT model by L1 group (means over 5 cross-validation folds).

	<u>Zero</u>	<u>Easy &gt; 0</u>	<u>Not Easy</u>	<u>(Very) Difficult</u>
CK	17,563	393	223	41
Non-CK	15,209	2,067	837	107

Table 4: Word counts in the whole dataset by L1 group and MAE tier.

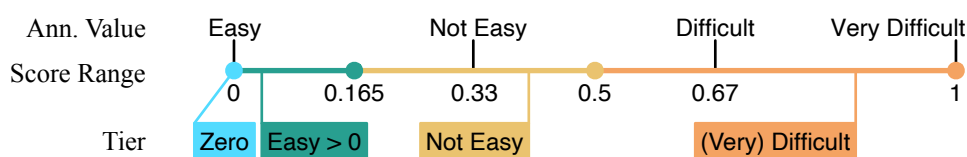


Figure 3: Illustrated score ranges of the MAE tiers:  $\{0\}$  for Zero,  $(0, 0.165]$  for Easy > 0,  $(0.165, 0.5]$  for Not Easy, and  $(0.5, 1]$  for (Very) Difficult.

tiers are centered at annotation values as illustrated in Figure 3. We handle zero as a special tier, and merge Very Difficult with Difficult due to a low number of words.

The fine-tuned BERT model for CK and Non-CK achieves  $R^2$  of 0.4351 and 0.6142, respectively. For both L1 groups, the MAE value increases markedly in each successive complexity tier, as the number of training examples (shown in Table 4) diminishes. Similarly, the CK model achieves lower error than non-CK only in tier Zero, where it has more examples available than the non-CK model. This suggests that the scarcity of words with complexity above zero is a factor contributing to worse performance on CK data, as measured by  $R^2$ .

## 6 Conclusion

In this paper, we presented the first dataset for Japanese LCP. It provides separate complexity scores based on the CK/non-CK distinction of annotators’ L1 background. Our analysis corroborates our conjecture that special consideration of L1 background is useful for the Japanese LCP task in particular. We believe it could benefit LCP in other languages as well.

In the baseline experiment, we demonstrated the efficacy of our BERT-based system for both CK and non-CK readers. Even after separating CK and non-

CK annotators, however, notable inter-annotator disagreement remains within these groups. Therefore personalized systems analogous to Gooding and Tragut (2022) could improve on our system. Future research should study this possibility, analyzing both its costs and benefits.

Models trained on JaLeCoN can be used as part of a lexical simplification pipeline for Japanese, both to identify complex words and to rank candidate simplifications. JaLeCoN itself can be further used as a basis for a lexical simplification dataset targeting words actually perceived as complex, similar to TSAR-ST datasets for English and Spanish (Štajner et al., 2022).

## Limitations

Our task setting and baseline system requires that the input is already segmented into words including MWEs. The MWE identification step in the construction process of our dataset involved time-consuming manual annotation. Building a high-quality system that fully automates the process is an issue for future work. Our dataset can be used to evaluate such a Japanese MWE identification system.

Additionally, as shown in Section 5, our baseline model performed relatively poorly in the higher complexity tiers. This is an effect of the dense annotation setting; it results in uneven distributions of

complexity as shown in Figure 2, where easy words greatly outnumber difficult words. One possible solution would be creating another LCP dataset using sparse annotation, where target words are selected using frequency bands so that the words are distributed across a wide range of frequency (Shardlow et al., 2022). Our data could provide insights as to what kind of words should be targeted by sparse annotation for such a dataset.

## Acknowledgments

We would like to express our gratitude to Justin Vasselli and the anonymous reviewers for their insightful feedback. This work was supported by JSPS KAKENHI grant number JP19K20351 and NAIST Foundation.

## References

- Yasuharu Den, Toshinobu Ogiso, Hideki Ogura, Atsushi Yamada, Nobuaki Menematsu, Kiyotaka Uchimoto, and Hanae Koiso. 2007. The development of an electronic dictionary for morphological analysis and its application to Japanese corpus linguistics. *Japanese Linguistics*, 22(5):101–123.
- 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.
- Sian Gooding and Ekaterina Kochmar. 2019. Complex word identification as a sequence labelling task. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1148–1153, Florence, Italy. Association for Computational Linguistics.
- Sian Gooding, Ekaterina Kochmar, Seid Muhie Yimam, and Chris Biemann. 2021. **Word complexity is in the eye of the beholder**. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4439–4449, Online. Association for Computational Linguistics.
- Sian Gooding and Manuel Tragut. 2022. **One size does not fit all: The case for personalised word complexity models**. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 353–365, Seattle, United States. Association for Computational Linguistics.
- Marcella Hu and I.S.P Nation. 2000. Unknown Vocabulary Density and Reading Comprehension. *Reading in a Foreign Language*, 13(1):403–30.
- Ekaterina Kochmar, Sian Gooding, and Matthew Shardlow. 2020. Detecting multiword expression type helps lexical complexity assessment. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4426–4435, Marseille, France. European Language Resources Association.
- Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Rebecca Knowles, Philipp Koehn, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Michal Novák, Martin Popel, and Maja Popović. 2022. Findings of the 2022 conference on machine translation (WMT22). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1–45, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.
- Keiko Koda. 1989. The effects of transferred vocabulary knowledge on the development of L2 reading proficiency. *Foreign Lang. Ann.*, 22(6):529–540.
- Kazuko Komori, Junko Mikuni, and Kondoh Atsuko. 2004. **Bunshō rikai o sokushin suru goi chishiki no ryōteki sokumen : Kichigo ritsu no ikichi tansaku no kokoromi [What percentage of known words in a text facilitates reading comprehension? : A Case Study for Exploration of the Threshold of Known Words] (in Japanese)**. *Nihongo Kyōiku [Journal of Japanese Language Teaching]*, 120:83–92.
- Shunsuke Kozawa, Uchimoto Kiyotaka, and Yasuharu Den. 2014. Adaptation of long-unit-word analysis system to different part-of-speech tagset. *Journal of Natural Language Processing*, 21(2):379–401.
- Klaus Krippendorff. 1970. **Bivariate Agreement Coefficients for Reliability of Data**. *Sociological Methodology*, 2:139–150.
- Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 230–237, Barcelona, Spain. Association for Computational Linguistics.
- Suguru Matsuyoshi, Satoshi Sato, and Takehito Utsuro. 2007. A dictionary of Japanese functional expressions with hierarchical organization. *Journal of Natural Language Processing*, 14(5):123–146.
- Daiki Nishihara and Tomoyuki Kajiwara. 2020. Word complexity estimation for Japanese lexical simplification. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3114–3120, Marseille, France. European Language Resources Association.
- Hideki Ogura, Hanae Koiso, Yumi Fujiike, Sayaka Miyauchi, Hikari Konishi, and Yutaka Hara. 2011. **Gendai nihongo kakikotoba kinkō kōpasu keitairon kiteishū dai 4 ban jō [Regulations of morphological information for balanced corpus of contemporary**

- written Japanese 4th edition volume 1] (in Japanese). *NINJAL Internal Reports*.
- Mai Omura, Aya Wakasa, and Masayuki Asahara. 2021. [Word delimitation issues in UD Japanese](#). In *Proceedings of the Fifth Workshop on Universal Dependencies (UDW, SyntaxFest 2021)*, pages 142–150, Sofia, Bulgaria. Association for Computational Linguistics.
- Paula J Schwanenflugel, Katherine Kip Harnishfeger, and Randall W Stowe. 1988. Context availability and lexical decisions for abstract and concrete words. *J. Mem. Lang.*, 27(5):499–520.
- Matthew Shardlow, Michael Cooper, and Marcos Zampieri. 2020. CompLex: A new corpus for lexical complexity prediction from Likert Scale data. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with READING Difficulties (READI)*, pages 57–62, Marseille, France. European Language Resources Association.
- Matthew Shardlow, Richard Evans, Gustavo Henrique Paetzold, and Marcos Zampieri. 2021. [SemEval-2021 task 1: Lexical complexity prediction](#). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1–16, Online. Association for Computational Linguistics.
- Matthew Shardlow, Richard Evans, and Marcos Zampieri. 2022. Predicting lexical complexity in english texts: the complex 2.0 dataset. *Language Resources and Evaluation*, 56(4):1153–1194.
- Sanja Štajner, Daniel Ferrés, Matthew Shardlow, Kai North, Marcos Zampieri, and Horacio Saggion. 2022. Lexical simplification benchmarks for english, portuguese, and spanish. *Front Artif Intell*, 5:991242.
- Yuriko Sunakawa, Jae-Ho Lee, and Mari Takahara. 2012. The construction of a database to support the compilation of japanese learners’ dictionaries. *Acta Linguistica Asiatica*, 2(2):97.
- Yuka Tateisi, Yoshihiko Ono, and Hisao Yamada. 1988. A computer readability formula of japanese texts for machine scoring. In *Coling Budapest 1988 Volume 2: International Conference on Computational Linguistics*.
- Yuki Taya, Lis Kanashiro Pereira, Fei Cheng, and Ichiro Kobayashi. 2021. [OCHADAI-KYOTO at SemEval-2021 task 1: Enhancing model generalization and robustness for lexical complexity prediction](#). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 17–23, Online. Association for Computational Linguistics.
- Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri. 2018. A report on the complex word identification shared task 2018. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 66–78, New Orleans, Louisiana. Association for Computational Linguistics.

## A Annotators

	Japanese		English	
	B1/B2	C1/C2	B1/B2	C1/C2
CK	4	3	7	0
Non-CK	6	1	2-3	4-5

Table 5: Annotator counts per sentence in each L1 group, by Japanese and English reading proficiency category. The proficiency levels were determined by self-reports with reference to an assessment grid either in Japanese<sup>15</sup> or in English.<sup>16</sup> Overall, our CK annotators are better at Japanese reading and poorer at English reading than the non-CK.

CK	Chinese: 6,	Korean: 1		
Non-CK	English: 2-3,	Thai: 2-3,	Indonesian: 1,	Lao: 1

Table 6: Annotator counts per sentence of each L1 group.

## B Complexity labels

3 (Very Difficult):	You hardly understand its meaning in the context.
2 (Difficult):	You can infer its meaning, but you are not confident.
1 (Not Easy):	You understand its meaning with confidence, but it is quite difficult among the expressions you can understand.
0 (Easy):	None of the above.

Table 7: Complexity labels. An annotator can label spans with complexity 3, 2, 1, or 0.

<sup>15</sup>[https://jfstandard.jp/pdf/self\\_assessment\\_jp.pdf](https://jfstandard.jp/pdf/self_assessment_jp.pdf)

<sup>16</sup><https://rm.coe.int/CoERMPublicCommonSearchServices/DisplayDCTMContent?documentId=090000168045bb52>

## C MWE Categories

Category	Description	Example
Lexicalized expressions	Non-compositional expressions whose meaning as a whole cannot be completely inferred by the meaning of their components.	使い   勝手 (ease of use)
Institutionalized expressions	Compositional expressions whose components cannot be replaced without distorting the meaning of the whole expression or violating the language conventions.	感染   症 (infectious disease)
Functional expressions	Expressions that behave like single function words.	に   つき   まし   て (as for)

Table 8: Categories we regard as MWEs. See [Kochmar et al. \(2020\)](#) for lexicalized and institutionalized expressions, and [Matsuyoshi et al. \(2007\)](#) for functional expressions. The vertical bars in the examples denote boundaries between SUWs.

## D Excluded categories

Category	Identification Approach	Example
Proper nouns	Proper nouns are first identified by MeCab. We also manually annotate proper noun phrases.	関東   大   震災 (The Great Kantō Earthquake)
Segmentation errors	We manually annotate sequences with segmentation errors.	も   や (mist)

Table 9: Categories of words or spans we exclude from our target. The vertical bars in the examples denote boundaries between SUWs. The correct segmentation for も | や is もや.

## E Distributions of Annotation

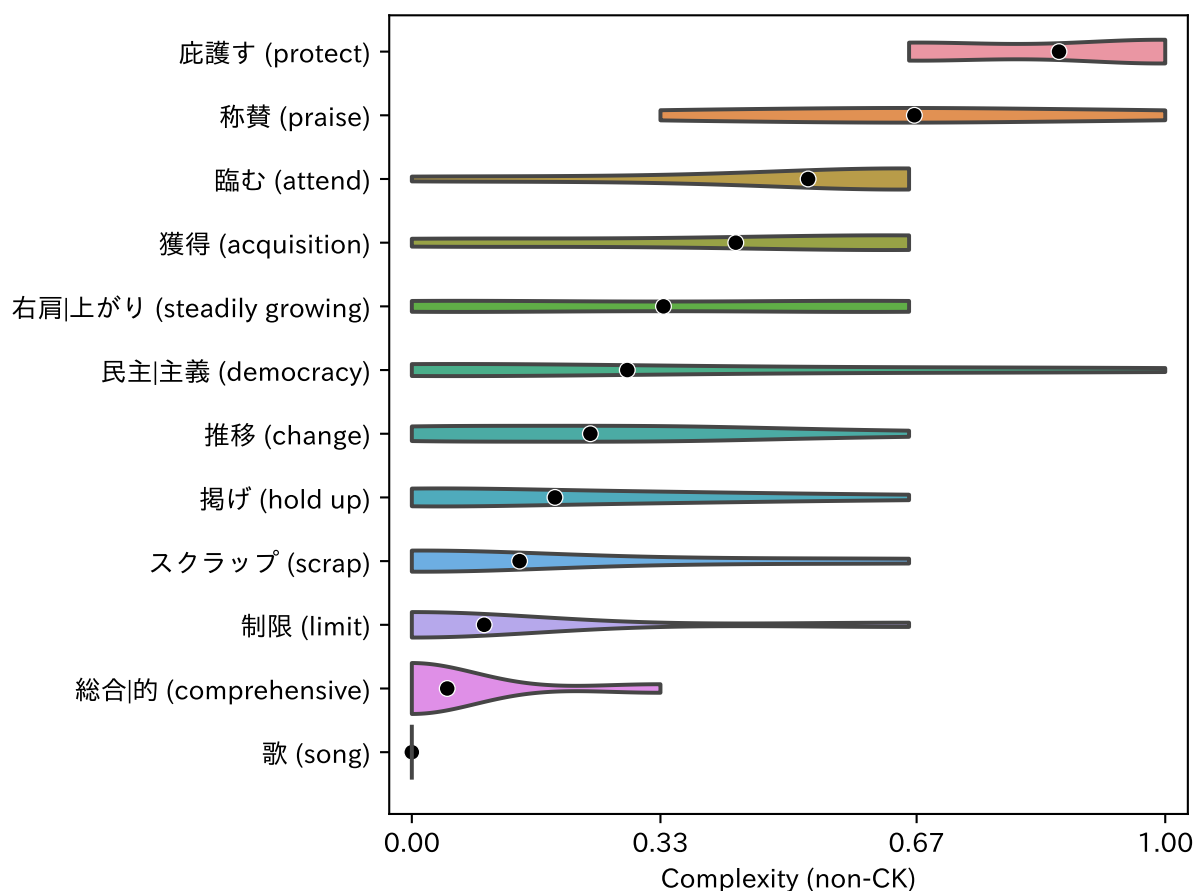


Figure 4: Violin plot showing annotation distributions of several words with dot markers showing the complexity scores, both for non-CK annotators. Words are shown in their surface forms; the vertical bars in them denote boundaries between SUWs.

## F Examples of Words by Origin

Origin	Containing Chinese characters ( <i>kanji</i> )?	
	Yes	No
Japanese ( <i>wago</i> )	歌 (song) 臨む (attend)	けれど も (although) ふさわしい (suitable)
Chinese/Sino-Japanese ( <i>kango</i> )	今回 (this time) 民主 主義 (democracy)	よう (it seems †様) もちろん (of course †勿論)
Other ( <i>gairaigo</i> )	旦那 (husband <Skt)	スクラップ (scrap <Eng) ホーム ページ (home page <Eng)

Table 10: Examples of words in JaLeCoN categorized by word origin and whether they contain Chinese characters. † marks a variant of the word written using Chinese characters documenting the Sino-Japanese origin; < marks the word's origin (Sanskrit or English). The vertical bars in the examples denote boundaries between SUWs. All categories except Other (*gairaigo*) written using Chinese characters are relatively common.

## G Experimental Setting

Optimizer	Adam ( $\beta_1 = 0.9, \beta_2 = 0.999$ )
– learning rate	5e-5
– schedule	no warm-up, linear decay
– L2 weight decay	0.01
Epochs	5
Loss function	Mean squared error
Dropout	0.1
Batch size	16
Weight initialization	$\mathcal{N}(\mu = 0, \sigma = 0.02)$ truncated to $\pm 2\sigma$
Bias initialization	0
Gradient L2 norm clipping	2

Table 11: Hyperparameters used for fine-tuning the BERT model. We have chosen the combination of learning rate (from 8e-6, 5e-5, 3e-5, and 2e-5), warm-up (from no warm-up and 10% steps), and the number of epochs (from 1 to 5) achieving the highest mean  $R^2$  in a nested 4-fold cross-validation on the training data of the first outer cross-validation split. The optimal combination was identical for CK and non-CK complexity.

Folds	5
Stratification	by genre (News and Government)
Grouping	by sequence of sentences (see Section 3)

Table 12: Cross-validation scheme used for fine-tuning and evaluation of the BERT model.

# Grammatical Error Correction for Sentence-level Assessment in Language Learning

Anisia Katinskaia<sup>†\*</sup> and Roman Yangarber<sup>†</sup>

University of Helsinki, Finland

\*Department of Computer Science

†Department of Digital Humanities

first.last@helsinki.fi

## Abstract

The paper presents experiments on using a Grammatical Error Correction (GEC) model to assess the correctness of answers that language learners give to grammar exercises. We empirically check the hypothesis that the GEC model corrects only errors and leaves correct answers unchanged. We perform a test on assessing learner answers in a real but constrained language-learning setup: the learners answer only fill-in-the-blank and multiple-choice exercises. For this purpose, we use ReLCo, a publicly available manually annotated learner dataset in Russian (Katinskaia et al., 2022). In this experiment, we fine-tune a large-scale T5 language model for the GEC task and estimate its performance on the RULEC-GEC dataset (Rozovskaya and Roth, 2019) to compare with top-performing models. We also release an updated version of the RULEC-GEC test set, manually checked by native speakers. Our analysis shows that the GEC model performs reasonably well in detecting erroneous answers to grammar exercises, and potentially can be used in a real learning setting for the best-performing error types. However, it struggles to assess answers which were tagged by human annotators as *alternative-correct* using the aforementioned hypothesis. This is in large part due to a still low recall in correcting errors, and the fact that the GEC model may modify even correct words—it may generate plausible alternatives, which are hard to evaluate against the gold-standard reference.

## 1 Introduction

Grammatical error correction (GEC) is the task of automatically detecting and correcting grammatical errors in text. Given the recent advancements in Transformer-based GEC models, which have the ability to suggest fluent and grammatically accurate corrections for input sentences, our focus lies in examining their application in language learning settings. One potential application is to check essays written by learners and provide suggestions for

corrections—this can be a useful tool for second-language (L2) learners to improve their writing. We are interested in incorporating GEC into an intelligent computer-aided language learning (CALL) system, but in a more constrained scenario: our objective is to evaluate whether a GEC model can be used for automatic assessment of the learner’s answers to fill-in-the-blank (“cloze”) and multiple-choice (MC) grammar exercises. We assume that this task is comparatively easier than correcting free-text essays, since the number of possible errors in each input sentence is constrained by the number of exercises, these exercises do not change the word order, and our focus is only on grammar. We empirically test the hypothesis: the GEC model can be employed to assess the grammatical correctness of learner answers to grammar exercises, because in an input sentence containing learner answers, the GEC model will fix only tokens with errors—for each erroneous answer it will suggest a correction, and will leave all correct answers unchanged.

In our setting, exercises are generated by *Revita*, a language learning system, which is used by several hundred L2 learners. These exercises are automatically generated based on a text selected for practice (Katinskaia et al., 2017, 2018). The system has one particular *expected answer* for each exercise—the one found in the original text. When doing an exercise, the learner may insert the expected answer, an error, or an *alternative-correct* answer, which is not expected, but fits the context. The problem can be stated as follows: an unexpected but suitable answer should be recognized as alternative correct, since providing incorrect feedback for valid answers can discourage learners (Katinskaia and Ivanova, 2019; Katinskaia and Yangarber, 2021). For example, in certain sentences, using the present or past tense can be equally acceptable. However, few corpora provide this type of annotation, therefore GEC models are predominantly trained and evaluated using only one reference per instance (Rozovskaya



and Roth, 2021; Bryant et al., 2022).

We use a freely available dataset [ReLCo](#), collected from Revita over several years ([Katinskaia et al., 2022](#)). This dataset contains short paragraphs with answers from learners of Russian. The paragraphs include *multiple* answers provided to the same grammar exercises, which were manually checked and tagged as acceptable or erroneous. To the best of our knowledge, this is the only freely available dataset of this type. As a GEC model, we fine-tune a pre-trained monolingual T5 language model ([Raffel et al., 2020](#)).

The contributions of this paper are: (1) We show that a GEC model can achieve reasonable performance in assessing erroneous answers for fill-in-the-blank and MC grammar exercises, if we use several top correction hypotheses. We empirically confirm the intuition that a Transformer-based GEC model *cannot* be used for assessing alternative-correct answers since top correction hypotheses can include corrections even for valid words. The lower-ranked hypotheses change the input sentence more freely: include more lexical changes, and more word removals or insertions. (2) We release a new version of the manually corrected [RULEC-GEC test set](#), which, we believe, can improve the evaluation of GEC models in the future. (3) We present the first experiment with [ReLCo](#) ([Katinskaia et al., 2022](#)), the semi-automatically collected learner data, to train a GEC model. Using [ReLCo](#) shows an improvement in GEC performance. (4) We extensively evaluate the performance of our model on the [RULEC-GEC](#) (henceforth—[RULEC](#)) test set automatically and manually, including an evaluation of several top hypotheses, and show an improvement of  $F_{0.5}$  score over the existing state-of-the-art results for Russian. Prior work showed that evaluating GEC output only by automatically comparing it with a single gold-standard reference per sentence results in *under-estimating* the performance ([Rozovskaya and Roth, 2021](#)).

The paper is organized as follows. Section 2 covers prior work on the GEC task. Section 3 describes the problem and our approach. Section 4 presents the data for training the GEC model, the training procedure, and the evaluation. Section 5 presents the experiments on assessing learner answers using the trained GEC model. It includes a discussion of results and error analyses. Section 6 presents the conclusions and future work.

## 2 Related Work

Most current approaches treat GEC as a natural language generation task. It can be formulated as a monolingual translation from incorrect to correct language using various architectures ([Yuan and Briscoe, 2016](#); [Junczys-Dowmunt et al., 2018](#); [Chollampatt and Ng, 2018](#); [Yuan et al., 2019](#); [Náplava and Straka, 2019](#); [Grundkiewicz et al., 2019](#); [Zhao et al., 2019](#); [Kaneko et al., 2020](#)). Due to the paucity of annotated training data for GEC, it has become standard practice to generate synthetic data, using various ways of creating erroneous sentences—by back-translation or random token-level transformations ([Kiyono et al., 2019](#)), using the history of Wikipedia edits ([Lichtarge et al., 2019](#)), confusion sets suggested by spell-checkers ([Grundkiewicz et al., 2019](#); [Náplava and Straka, 2019](#)), real error patterns ([Choe et al., 2019](#); [Takahashi et al., 2020](#); [Li and He, 2021](#); [Stahlberg and Kumar, 2021](#)), or applying noise to a latent representation of an error-free sentence ([Wan et al., 2020](#)). A comparative study of methods of generating synthetic data is presented in ([White and Rozovskaya, 2020](#)).

Another approach is text editing—generating a sequence of edits to apply to the incorrect input sentence ([Malmi et al., 2019](#); [Stahlberg and Kumar, 2020](#); [Tarnavskiy et al., 2022](#)). In [GEC-TOR](#) ([Omelianchuk et al., 2020](#)), the authors develop a set of custom token-level transformations to recover the target text from the source. Editing is faster than generating the whole corrected sentence, but requires constructing many language-specific transformations. More on GEC and existing approaches to the problems and evaluation is reviewed in ([Bryant et al., 2022](#)).

A number of papers focus on the actual use of GEC models by language learners. [Homma and Komachi \(2020\)](#) approach the problem of GEC usability as a part of a writing-support system for Japanese, with a focus on inference speed and working with incomplete sentences. [Zomer and Frankenberg-Garcia \(2021\)](#) present a writing-improvement model, which is adapted to the writer’s first language (L1). The model’s output was evaluated on grammaticality, acceptability, and lexical and syntactic diversity. An Example-Based GEC with a focus on interpretability is introduced in ([Kaneko et al., 2022](#)): the model presents to the learners correction results and examples as a base for correction. [Takahashi et al. \(2022\)](#) explore the learners’ proficiency-wise evaluation for Quality

Exercise: Вероятно, такие приборы  
преборы уже изобрести .

Answer: Вероятно, такие **преборы** уже **изобрели**.



1. Вероятно, такие **приборы** уже **изобрели**.
2. Вероятно, что такие **приборы** уже **изобрели**.
3. Весьма вероятно, такие **преборы** уже **изобрели**.

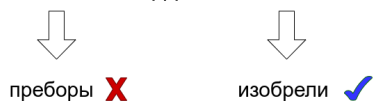


Figure 1: Proposal for how to use a GEC model to check the learner’s answers in automatically generated exercises: if an answer was corrected in the *majority* of the top-7 hypotheses, assume it is an error. Otherwise, assume it is correct if it was not altered in the top-3 hypotheses. Red denotes incorrect forms, blue—correct.

### Estimation (QE) of GEC.

Several papers on GEC focus on low-resource languages, including Russian (Rozovskaya and Roth, 2019; Katsumata and Komachi, 2020). Náplava and Straka (2019) adapted the approach of Grundkiewicz et al. (2019) for Russian, German, and Czech. Their results on Russian outperformed those of Rozovskaya and Roth (2019) by more than 100% on the  $F_{0.5}$  score, but still performed quite poorly compared with other languages. GEC for Russian is shown to be a challenging task, which is explained by the small size of the RULEC corpus. In (Rothe et al., 2021), the biggest multilingual T5 model which was pre-trained on synthetic data and fine-tuned on real data achieved the best performance on Russian among other approaches. Performance was improved by adding to the GEC pipeline a Transformer model for re-ranking the suggested correction edits (Sorokin, 2022).

## 3 Problem Setup

Our task is to evaluate and provide feedback on the grammatical correctness of answers given by the learner to all grammar exercises (cloze and MC) generated by a CALL system in a sentence. For example, in the sentence in Figure 1, “Вероятно, такие **приборы** уже **изобретены**.” (“Probably, such **gadgets** have already **been invented**.”), the learner receives one MC exercise (*приборы* vs. *преборы*, “gadgets”) and a cloze exercise with lemma “изобрести” (“to invent”). The MC has only one correct answer, if the exercise is well-designed. In the cloze, the student’s an-

Dataset	Training	Develop	Test
RULEC	4 980	2 500	5 000
cLang-8 (Ru)	44 830	—	—
ReLCo	8 560	—	7 017

Table 1: Counts of sentence pairs in annotated datasets.

swer can be: (1) a definite error; (2) definitely correct, if it matches the expected past passive form “*изобретены*”; or (3) impersonal past tense “*изобрели*”, which is an *acceptable*, slightly different way of saying the same thing (“*Probably someone has already invented such gadgets*”). These alternative corrections can be potentially incorrect in a wider context, but we focus on the context of one sentence to simplify the task.

The proposed approach is to use a GEC model whose input is a sentence with all of the learner’s answers inserted jointly. This is important, because words chosen by the CALL system for exercises can grammatically depend on each other, and various combinations of answers could be correct, e.g., “gadgets have” vs. “a gadget has.” Our conjecture is: if an answer was corrected by a GEC model, it is likely an error; if it was not corrected, it is probably correct. To increase our trust in the model’s predictions and address the potential issue of under-corrected errors, we employ a beam search to generate multiple top-ranked hypotheses instead of relying solely on the top-1 correction, see details in Section 5. Previous research by Rozovskaya and Roth (2021) has demonstrated experimentally that lower-ranked hypotheses produced by GEC systems could also be taken into account because they can be qualitatively even better than the top-1 hypotheses, which often suffer from the tendency of GEC systems to under-correct errors due to training with one gold reference per input sentences.

## 4 GEC Experiments

### 4.1 Data

To train the GEC model, we use the datasets: RULEC (Rozovskaya and Roth, 2019), Russian cLang-8 (Rothe et al., 2021), and ReLCo (Katin-skaia et al., 2022). Dataset statistics are in Table 1. Incorporating the Lang-8 (Tajiri et al., 2012) corpus did not yield a significant improvement, see Table 3. Similar results were shown by Trinh and Rozovskaya (2021), where adding RU-Lang8 to the training data did not improve the results on the

Split	# Errors	# Alternative-Correct
Train	5 642	418
Test	4 316	1 289

Table 2: Number of answers which were manually annotated in ReLCo. Right column: AC learner answers—manually tagged by annotators as “correct”, but differ from the expected “reference” answers. Center column: answers which were manually tagged as “errors”.

RULEC test either, although their experiments were conducted using a different model. Therefore, we included the Russian part of the cLang-8 dataset, which is a cleaned version of Lang-8. cLang8 was used only for training. For tuning parameters and analysis of model outputs, we use only the RULEC validation set. The RULEC test set was used for evaluation and comparison with other GEC models.

We split the manually annotated ReLCo into a training and test set. ReLCo consists of short paragraphs, which include learner answers given to grammar exercises. Exercises in the same paragraph can vary depending on the learner’s proficiency. The same paragraphs can be practiced by different students or by the same student multiple times, resulting in numerous repeating sentences in the corpus. We ensured that the same sentence never occurs in different data splits. Since we are interested in GEC performance on sentences with multiple acceptable corrections—henceforth, *alternative correct*, or AC—we placed more of such sentences into the test set, see the number of erroneous and AC answers in each data split in Table 2. We also do not want the GEC model to be forced to replace AC answers with expected answers during fine-tuning.

## 4.2 GEC Model

We use the Text-to-Text Transfer Transformer (T5) model, an encoder-decoder multi-task model that was pre-trained on unsupervised and supervised tasks, with converting each task into a text-to-text format. Rather than the multilingual T5 as in Rothe et al. (2021), we fine-tuned a monolingual Russian T5 model (Raffel et al., 2020).

Rothe et al. (2021) showed that bigger T5 models perform GEC better for all tested languages. We chose a large-size configuration (over 700M parameters), since we cannot run T5 xl or T5 xxl with available resources.

The T5 model is instructed to perform a par-

Model	Training Data	$F_{0.5}$
ruT5 large	RULEC	38.10
ruT5 large	RULEC + Lang-8	38.90
ruT5 large	RULEC + cLang-8	39.50
ruT5 large	RULEC + cLang-8 + ReLCo	43.74

Table 3:  $F_{0.5}$  scores on the RULEC test data calculated with  $M^2$  scorer. All T5 models reported in this table are not pre-trained on synthetic data.

ticular generation task by adding a prefix at the beginning of an input sequence. We conditioned each input sentence by adding the task definition “improve\_grammar”.<sup>1</sup> First, we tried to directly fine-tune the T5 model on the real data, see the results of tuning with several combinations of learner corpora in Table 3. The combination of the RULEC train partition, cLang8, and the ReLCo train partition yields the best F-score and, therefore, it was used in all the following experiments.

Since Rothe et al. (2021) report that the best-performing setup for the T5 model used GEC pre-training on synthetic data, we also (1) pre-train the T5 model on a synthetic dataset until convergence,<sup>2</sup> followed by (2) fine-tuning on the three mentioned datasets.<sup>3</sup> The synthetic data was generated from WMT News Crawl monolingual training data (Bojar et al., 2017) using Aspell confusion sets following (Grundkiewicz et al., 2019). We generated 10M sentences using the same parameters as presented in (Náplava and Straka, 2019). To choose parameters for the fine-tuning on original data, we run hyper-parameter search<sup>4</sup> using Population Based Training (PBT) optimization algorithm (Jaderberg et al., 2017). We set the dropout rate of the T5 model at 0.2, which was found to give the biggest gain in F-score on a validation set. Higher dropout may teach the model to trust the source sentence less and introduce more corrections, as noted previously by Junczys-Dowmunt et al. (2018).

## 4.3 GEC Evaluation

Given an original sentence with errors (*a source sentence*), the GEC system generates a ranked list of suggested corrections (*hypotheses*). The perfor-

<sup>1</sup>Our implementation is based on Hugging Face.

<sup>2</sup>3 GPU V100, pre-training for 1.48M steps with batch size = 6, weight decay = 0, learning rate = 5e-5.

<sup>3</sup>The fine-tuned model is available at [RuT5\\_GEC](#)

<sup>4</sup>The best performance was obtained with the following parameters: number of epochs = 2, weight decay = 0.180335, learning rate = 3.83229e-05.

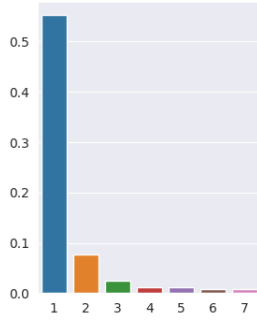


Figure 2: Percentage of hypotheses equal to the gold references in the test set (y-axis) by the rank of the hypotheses (x-label).

mance that we discuss next is calculated for the top-1 hypotheses.

**Evaluation with  $M^2$  Scorer:** Evaluation of all GEC models was done on the RULEC test set using the MaxMatch ( $M^2$ ) scorer (Dahlmeier and Ng, 2012). It computes GEC performance in terms of phrase-level edits. The results of the evaluation and effect of pre-training and tuning hyper-parameters are shown in Table 5; all reported scores are averaged over 3 runs. A simple pre-processing improvement of the data<sup>5</sup> gives a performance gain, see “data preprocessing” in Table 5. We also have detected some formatting issues and word repetitions in the hypotheses generated by the model for the validation set. Therefore, we run post-processing for the model output, see details in Appendix A. The results of the evaluation after post-processing are in Table 4 and Table 5, and they are on par with the current state of the art.

**Evaluation with ERRANT.** ERRANT (Bryant et al., 2017; Felice et al., 2016) is a reference-based scorer which measures performance in terms of an edit-based F-score. Unlike the  $M^2$  scorer, it is also able to calculate error type scores at different granularity, e.g., *Replacement* edit or *Replacement:Noun:Case* edit. We use an extension<sup>6</sup> of ERRANT for Russian (Katinskaia et al., 2022). Evaluation of GEC performance using ERRANT was not reported for Russian in the previously published papers. We measured performance with ERRANT on

<sup>5</sup>Inspection of the GEC model’s output on the validation set showed that during inference, the pre-trained and fine-tuned T5 model inserts white spaces into tokens containing the characters [´] (“stress”), or [ë].

Filtering out stress characters and replacing [ë] with [e] is a trivial fix that does not alter the meaning of the text.

<sup>6</sup>RuERRANT

System	$P$	$R$	$F_{0.5}$
Rozovskaya and Roth (2019)	38.0	7.5	21.0
Trinh and Rozovskaya (2021)	59.1	26.1	47.2
Náplava and Straka (2019)	63.3	27.5	50.2
Rothe, Mallinson, Malmi, Krause, and Severyn (2021) mT5 large	-	-	27.6
Rothe, Mallinson, Malmi, Krause, and Severyn (2021) gT5 xxl	-	-	51.6
Our model	66.6	29.1	52.9

Table 4: Performance of different GEC models for Russian, calculated using  $M^2$  scorer on the RULEC test set.

Model	$F_{0.5}$
ruT5 + RULEC + cLang-8 + ReLCo	43.74
large + synthetic pre-training	49.62
+ tuned hyper-parameters	50.83
+ data preprocessing	51.82
+ output post-processing	52.94
+ tested on re-annotated RULEC	55.35
+ COMET re-ranking	68.19

Table 5:  $F_{0.5}$  scores on the RULEC test data calculated using  $M^2$  scorer.

the post-processed output of the best GEC model which we trained, see Table 6. ERRANT’s  $F_{0.5}$  score for correction is lower (52.1) than  $F_{0.5}$  calculated by  $M^2$  scorer (52.9). The same discrepancy between these scores was reported in (Kiyono et al., 2019) for English.

The T5-based GEC model is performing significantly better for replacement errors than for insertion or deletion errors. One possible reason for that can be related to the distribution of error types in the training data: syntactic data was generated mostly by replacing tokens; and in the real learner data, errors were corrected following the principle of minimum correction needed to fix the source sentence, which mostly involves replacing separate words rather than removing or inserting words. ReLCo includes only replacement errors collected from cloze and MC exercises.

**Manual Evaluation.** Of the total 5 000 top-1 GEC hypotheses generated for the test set, 1 199 were found to be different from both the source sentences and the corresponding gold-standard references. These hypotheses were manually evaluated

Error type	<i>P</i>	<i>R</i>	$F_{0.5}$	<i>P</i>	<i>R</i>	$F_{0.5}$
	<i>Detection</i>			<i>Correction</i>		
Insertion	38.5	8.9	23.2	32.6	7.6	19.6
Replacement	80.9	41.5	67.9	69.6	35.8	58.5
Deletion	36.4	5.3	16.7	24.0	3.2	10.3
Overall	76.0	33.5	60.6	65.3	28.7	52.1

Table 6: Precision, Recall, and F-score measured by ERRANT for span-based error *detection* (left) and span-based error *correction* (right). “Overall” shows performance on all three types of error edits.

by a native-speaking annotator with a degree in teaching Russian and prior annotation experience. The task was to mark whether a sentence is acceptable grammatically. The results showed that 285 of the checked hypotheses can be considered grammatically acceptable. In some cases, the corresponding gold references include typos or uncorrected errors, while in others, GEC hypotheses and the gold reference both present alternative corrections of the source sentence. In addition, 52 hypotheses differ from their gold references only by capitalization, e.g., the first word is not capitalized in a reference, but it is capitalized in the generated hypothesis. The remaining 862 sentences were annotated as indeed ungrammatical. As a final result, the manual evaluation showed that 62.4% of all 5 000 suggested top-1 hypotheses are correct.

**Other Hypotheses.** We generated 7 hypotheses<sup>7</sup> for each source sentence with beam search decoding. Comparing hypotheses with the references shows that in some cases the GEC model produces a correction which is the same as the reference sentence, but it is not chosen as the top-1 hypothesis. Figure 2 shows the percentage of hypotheses equal to the reference sentences by the rank of the hypotheses. Ranked top 3 include 65.5% of hypotheses equal to the references. More on the manual evaluation of the top-3 hypotheses is in Appendix B.

#### 4.4 RULEC Test Cleaning

Testing various models on the RULEC test set showed that it contains uncorrected errors, ungrammatical corrections, and mistakes in indexing of proposed corrections. Since this impedes assessing the true performance of the models, we undertook a re-annotation of the data. At this stage, we do

<sup>7</sup>This number of hypotheses is the maximum we can generate with resources available to us.

not claim that all errors and inconsistencies in the RULEC test set have been fixed.

Annotation was done by three native speakers: two Master’s students in linguistics and one expert in teaching Russian. Source sentences were randomly split into two subsets and presented to two annotators, 2.5K sentences each. The annotators could see the original erroneous sentence and its correction (gold reference) proposed in RULEC. The task was to fix the gold reference only if needed, following the minimal-edits principle that results in a grammatically correct reference sentence. The third annotator checked all 5K source sentences and the proposed corrections. Due to limited resources, we could not involve more annotators to correct source sentences without seeing the gold references, or to get more corrections per source sentence. We measured the agreement between the last annotator and the two annotators in the first phase of correction: average agreement is 87%. Most disagreements relate to punctuation, and were resolved by the final annotator.

We calculated our GEC model performance (with output post-processing) on the corrected RULEC test set, see the last row in Table 5. The  $F_{0.5}$  score increases to 55.4, which is above the current state-of-the-art results for Russian. A re-annotated test set allowed us to evaluate more realistically the corrections which were attempted by the model, though many errors are still left uncorrected. The updated test set is released in  $M^2$  format.<sup>8</sup>

## 5 GEC for Evaluating Learner Answers

The following task is to evaluate whether a GEC model can be directly used for assessing learners’ answers in a CALL system.

**Evaluation.** We generated 7 hypotheses for each sentence in the ReLCo test set. The source sentences did not need pre-processing. We applied a post-processing step to filter out 874 hypotheses containing word repetitions, following the method used for the RULEC test set: a filtered-out hypothesis is replaced with its source sentence.

Next, we describe the procedure for checking learner answers based on the suggested corrections. We define the word inserted by the learner as an answer to an exercise as the *target* word. Firstly, we align the suggested GEC hypotheses with the corresponding source sentences. Then, for each

<sup>8</sup>RULEC-GEC test updated

Answer type	# of answers	$P$	$R$	$F_{0.5}$	$F_1$	Acc.	$P$	$R$	$F_{0.5}$	$F_1$	Acc.	$P$	$R$	$F_{0.5}$	$F_1$	Acc.	
				<i>top-3</i>					<i>all</i>				<i>top-3 &amp; re-ranked</i>				
Gram. error	4 316	89.5	81.9	<b>87.7</b>	83.7	-	87.0	87.8	87.1	87.6	-	82.9	90.9	84.4	<b>88.8</b>	-	
AC	1 289	52.4	67.6	54.8	63.0	-	57.2	55.6	56.9	55.9	-	55.5	72.1	<b>58.2</b>	<b>67.1</b>	-	
Hard AC	206	-	-	-	-	55.3	-	-	-	-	40.8	-	-	-	-	<b>59.2</b>	

Table 7: Results of estimating the correctness of learners’ answers using GEC hypotheses. AC denotes answers which were manually tagged as correct. Hard AC denotes AC answers with the highest disagreement rate among annotators, performance score is accuracy because all instances belong to one class. The best scores are in bold. *top-3*—an answer is considered correct if it is unchanged in all top 3 hypotheses; *all*—an answer is unchanged in all 7 hypotheses; *top-3 & re-ranked*—an answer is unchanged in top-3 hypotheses after re-ranking with COMET score.

target word, we follow the steps:

1. Check whether a target word was corrected by the *majority* of the suggested hypotheses.
2. If corrected by majority, the target word is classified as a grammatical error.
3. Otherwise, check whether the target word was left unchanged in *all* top 3 hypotheses.
4. If not corrected in all top 3 hypotheses, it is potentially an alternative correct answer.
5. Else it is classified as an error.

We chose to evaluate the top-3 hypotheses because previous testing on RULEC showed that they had the highest quality among all generated hypotheses. The results of evaluating grammatical correctness of answers using this algorithm are presented in Table 7, see the third column marked “*top-3*”. Besides  $F_{0.5}$ , we report the  $F_1$  score: for language learning, it is important not only to provide valid corrections (low false positives) but also not to silently miss errors (low false negatives). Examining multiple hypotheses allows us to improve the precision of detecting AC and the recall of detecting errors. We experimented with modifying steps (3) and (4) by requiring the target word to remain unchanged in all seven suggested hypotheses (see column “*all*” in Table 7). Furthermore, we compared performance on AC answers with the highest disagreement rate among annotators, referred to as “Hard AC” in Table 7. The table presents the performance measured in accuracy, which indicates how many Hard AC answers are recognised as correct.

**Re-ranking.** One of the problems with using all hypotheses, or only the top- $N$ , is that some of the hypotheses can include more uncorrected errors or can differ significantly from the source sentence lexically and syntactically. For this reason, we experiment with several methods for scoring and re-ranking hypotheses, e.g., using LM

scores, the number of errors detected by a GED model, VERNet (Liu et al., 2021), Discriminative re-ranking (Lee et al., 2021), OpenAI’s GPT-3.5 model<sup>9</sup> as re-ranker, etc. We test them on RULEC and choose COMET as the best-performing score. Different methods allow increasing precision or recall which, depending on the use case, can be beneficial.

COMET metric<sup>10</sup> for MT evaluation (Stewart et al., 2020; Rei et al., 2020) exploits information from both the source sentence and the reference in order to evaluate the quality of an MT hypothesis. Unlike re-ranking methods which are not using any information about references, COMET allowed to get significant improvement, see performance of a GEC model evaluated after re-ranking with COMET in Table 5. This is the first application of this metric to GEC.

Table 7 (column “*top-3 & re-ranked*”) shows results of assessing learner answers after re-ranking hypotheses according to their COMET score. Using top-3 hypotheses and re-ranking shows the best scores for assessing learners answers overall.

## 5.1 Error Analysis

We separately analyzed the GEC model’s performance in assessing alternative correct answers and erroneous answers.

**Alternative Correct (AC).** Table 8 shows the accuracy of the GEC model on 14 different types of AC answers in the test data, which were annotated manually. The notation “*Tense: past/present*” means that the expected answer was in the past tense, the learner’s answer was in the present, and both forms are acceptable in the context. Performance significantly varies across different types, which should be considered when utilizing GEC

<sup>9</sup><https://platform.openai.com/docs/models/gpt-3-5>

<sup>10</sup>The model used is Unbabel/wmt22-comet-da

AC category	%	AC category	%
Tense: past/present	85.0	Tense: past/fut.	56.3
Preposition	70.8	Verb: transgr./past	55.6
Number: plur./sing.	68.2	Case: gen./accus.	52.9
Number: sing./plur.	67.2	Adj.: short/full	52.5
Tense: present/past	66.9	Aspect: perf./imperf.	48.7
Tense: fut./past	66.7	Case: instruct./nom.	33.3
Aspect: imperf./perf.	66.4	Case: accus./loc.	31.5

Table 8: Accuracy on estimating AC answers by the GEC model for different categories. Notation “past/pres.” means that the learner replaced the past tense form with the present tense; “transgr.” denotes transgressive.

for assessing learner answers.

We found that sometimes the GEC model proposes to correct AC answers by words with similar spelling but different meaning that are not relevant in the context, e.g., “бесплотны” (“*ethereal*”) is corrected as “бесплатны” (“*free of charge*”); “от вора” (“*from a thief*”) is corrected as “от ворот” (“*from the gate*”). It especially relates to rare words, e.g., “калорифер” (“*heater*”) replaced with “калории” (“*calories*”). In many cases, the GEC model indeed does not change an AC answer, but it frequently proposes the expected correct answer as a correction, e.g., the top-2 suggestions (Output 1 and 2) in Table 9 include both “смотри” and “посмотри”.<sup>11</sup> For more examples, see Appendix D.

Potentially, GEC may be used only for the best-performing types, while for other types, we might need to train separate models. We could provide a learner with 2-3 top corrections suggested by the model and, if it is possible, involve a teacher in a final assessment step.

**Errors.** One of the detected problems relates to a mismatch between annotation and evaluation conditions: learners’ answers in ReLCo were annotated within the context of a paragraph, while we have run GEC evaluation of separate sentences. Therefore, some answers, which are erroneous within a paragraph but not a sentence, were not detected as errors by the model. We run a preliminary evaluation by providing the model with whole paragraphs as input, instead of sentences. Some longer paragraphs have to be pruned to 100 tokens.<sup>12</sup> Performance drops in terms of recall for error detection, though precision increases, especially for a setting

<sup>11</sup>“Look” in imperfect and perfect aspect, respectively.

<sup>12</sup>Due to technical limitations, the input sentence length for beam search cannot exceed 100 tokens.

<b>Source:</b> на <u>привокзальных</u> площади
<b>Output:</b> на <u>привокзальных</u> площадях
<b>Expected:</b> на <u>привокзальной</u> площади ( <i>at station square</i> )
<b>Source:</b> Он <u>из-за</u> <u>этих</u> <u>документы</u> отвечает.
<b>Output:</b> Он <u>из-за</u> <u>этих</u> <u>документов</u> отвечает.
<b>Expected:</b> Он за эти документы отвечает. ( <i>He is responsible for these documents.</i> )
<b>Source:</b> <u>во</u> перерыве между забегами
<b>Output:</b> во время <u>перевыва</u> между забегами
<b>Expected:</b> <u>в</u> перерыве между забегами ( <i>during the break between runs</i> )
<b>Source:</b> Да ты под переплетом <u>смотри</u>
<b>Output 1:</b> Да ты под переплетом <u>посмотри</u>
<b>Output 2:</b> Да ты под переплетом <u>смотри</u>
<b>Expected:</b> Да ты под переплетом <u>посмотри</u> ( <i>Why don't you look under the book cover</i> )

Table 9: Examples of some source phrases (“Source”) with learners’ answers (underlined) which were corrected by the model (“Output”). “Expected” shows which answers were expected by Revita CALL system. Red denotes incorrect answers, blue—correct.

with re-ranking. See more details in Appendix C. Paragraph-level assessment needs more investigation in future work.

Another issue is that the GEC model is not informed which word is a target of an exercise. In the first example in Table 9, only the underlined word “привокзальных” (“*near railway station*”) was provided as an answer which is an incorrect plural form in the context. However, the model corrected the noun “площади” (“*square*”) from singular to a plural form. The second example shows issues with reverted word order: the model detects local syntactic relations between the preposition “из-за” (“*because of*”) and the following noun phrase “этих документов” (“*these documents*”), so it puts the noun phrase in genitive case. However, it failed to detect government relations with a head verb “отвечает” (“*is responsible*”), which requires the preposition “за” (“*for*”), not “из-за”. The last example shows an issue with checking whether an answer was corrected by the majority of the hypotheses: instead of correcting a preposition “в” (“*in*”), the model rephrases the whole time expression.

Figure 3 shows the evaluation of detection and correction performance for several error types using ERRANT, on the RULEC and ReLCo test sets. Performance on the two test sets differs drastically on some error types: e.g., spelling, verb aspect, and

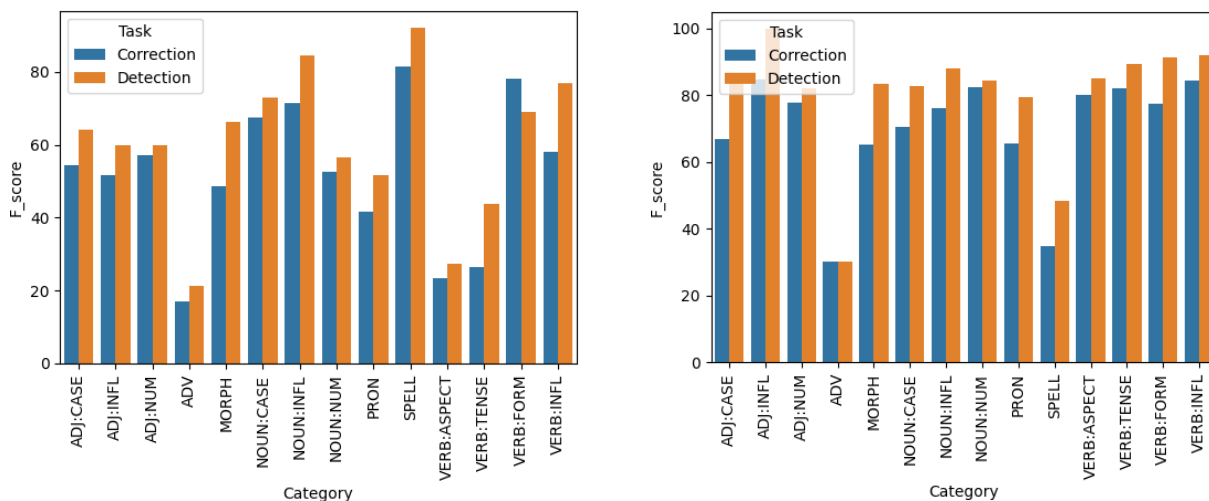


Figure 3: Performance of the GEC model in terms of  $F_{0.5}$  score for different error types in the RULEC test set (left) and the ReLCo test set (right).

tense errors. Adverbs and pronouns have low performance in both test sets. All scores for ReLCo are higher, likely because it includes only replacement errors that are better handled by the GEC model. This indicates that the model can potentially be used for detecting errors in cloze exercises with best-performing error types, without providing suggested corrections, since correction performance is lower.

## 6 Conclusions and Future Work

We present experiments on using Transformer-based GEC models to evaluate the correctness of answers provided by language learners to grammar exercises. To the best of our knowledge, it is the first attempt to directly employ a GEC model for this task. We find that the top-performing GEC model demonstrates the potential to detect and correct errors in user answers provided to fill-in-the-blank and multiple-choice grammar exercises, if we use multiple top hypotheses generated with beam search. However, this approach is less effective for assessing alternative-correct answers. Given the current low recall of the GEC model, there is a high chance of labeling erroneous answers as acceptable. Furthermore, the number of possible alternative corrections proposed by more advanced GEC models can be high, meaning that when the GEC model corrects an answer, it does not necessarily indicate the presence of an error.

The problem of evaluating alternative correct answers is equivalent to the problem of multiple possible corrections for a given error span in GEC. This issue is particularly challenging because GEC

models are primarily trained and evaluated using a single reference for each sentence, as discussed in (Rozovskaya and Roth, 2021; Bryant et al., 2022). In our future work, we aim to focus on developing methods for evaluating the suggested corrections by combining reference-based and reference-free scoring approaches.

While GEC is typically approached as a task involving isolated sentences, there have been studies addressing document-level GEC as well (Chollamatt et al., 2019; Yuan and Bryant, 2021). In our experiments, we also focused on assessing grammatical correctness at the sentence level. However, in future work, we plan to investigate the assessment of learner answers within a paragraph. We intend to conduct further research on leveraging large language models to evaluate the acceptability of answers and explore the combination of various re-ranking methods.

## Acknowledgements

This work was supported in part by the Academy of Finland, Helsinki Institute for Information Technology (HIIT), BusinessFinland (Grant “Revita”, 42560/31/2020), and Tulevaisuusrahasto, the Future Development Fund, Faculty of Arts, University of Helsinki.

## 7 Ethical Considerations

We use only publicly available resources for all conducted experiments. All annotators were volunteer students who performed the tasks as a part of their studies and received credits for it.



## 8 Limitations

The current work has a number of limitations to consider.

(A) The paper’s experimental design was limited to a single language because we are not aware of any other learner corpora with multiple answers provided to the same exercises.

(B) The described approach to assessing the correctness of learner answers is limited by its design. First, the number of GEC hypotheses to check depends on the GEC model’s performance and, potentially, on the language. Second, if a word was not corrected, it can be a false negative error instead of a correct answer. Third, the GEC model can suggest corrections (valid and not valid) even to a correct answer depending on the data it was trained on.

(C) Our approach focuses only on grammatical errors and it does not take into account semantic or pragmatic errors.

(D) Due to limited resources, we were unable to involve more people with prior annotation experience in the re-annotation of the RULEC test set, as well as in the manual verification of hypotheses generated by the GEC model. We acknowledge that the annotation performed by our annotators may not be entirely error-free: the annotators were free to work at their own pace and therefore could potentially rush and make errors themselves. Hence, we do not claim that the re-annotated RULEC test set does not include any inconsistency anymore. We believe that the existing datasets should be thoroughly checked, given the small amount of learner data available for languages other than English, before utilizing them to train and evaluate new models.

(E) Considering the practical use of our GEC model as a component of a CALL system, we find that it can potentially be used in a limited context, i.e., for checking answers provided to cloze and multiple-choice exercises, only for best-performing error types. As for alternative correct answers, even for best-performing categories of answers, a human teacher should verify proposed corrections. We have to underline that learner errors in RULEC-GEC (and especially in any synthetic dataset) can significantly differ from errors made by learners with various backgrounds, native languages, and proficiency levels. We also find that low recall of state-of-the-art GEC models impedes their usage in language learning settings. At the moment, learner answers should be verified by a human teacher.

## References

- Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. [Findings of the 2017 conference on machine translation \(WMT17\)](#). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.
- Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. [Automatic annotation and evaluation of error types for grammatical error correction](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.
- Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. 2022. [Grammatical error correction: a survey of the state of the art](#). *arXiv preprint arXiv:2211.05166*.
- Yo Joong Choe, Jiyeon Ham, Kyubong Park, and Yeoil Yoon. 2019. [A neural grammatical error correction system built on better pre-training and sequential transfer learning](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 213–227, Florence, Italy. Association for Computational Linguistics.
- Shamil Chollampatt and Hwee Tou Ng. 2018. [A multi-layer convolutional encoder-decoder neural network for grammatical error correction](#). In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*.
- Shamil Chollampatt, Weiqi Wang, and Hwee Tou Ng. 2019. [Cross-sentence grammatical error correction](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 435–445, Florence, Italy. Association for Computational Linguistics.
- Daniel Dahlmeier and Hwee Tou Ng. 2012. [Better evaluation for grammatical error correction](#). In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572, Montréal, Canada. Association for Computational Linguistics.
- Mariano Felice, Christopher Bryant, and Ted Briscoe. 2016. [Automatic extraction of learner errors in ESL sentences using linguistically enhanced alignments](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 825–835, Osaka, Japan. The COLING 2016 Organizing Committee.
- Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. [Neural grammatical error correction systems with unsupervised pre-training](#)

- on synthetic data. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263, Florence, Italy. Association for Computational Linguistics.
- Hiroki Homma and Mamoru Komachi. 2020. Non-autoregressive grammatical error correction toward a writing support system. In *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 1–10, Suzhou, China. Association for Computational Linguistics.
- Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, et al. 2017. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*.
- Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. Approaching neural grammatical error correction as a low-resource machine translation task. 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 595–606, New Orleans, Louisiana. Association for Computational Linguistics.
- Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020. Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4248–4254, Online. Association for Computational Linguistics.
- Masahiro Kaneko, Sho Takase, Ayana Niwa, and Naoaki Okazaki. 2022. Interpretability for language learners using example-based grammatical error correction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7176–7187, Dublin, Ireland. Association for Computational Linguistics.
- Anisia Katinskaia and Sardana Ivanova. 2019. Multiple admissibility: Judging grammaticality using unlabeled data in language learning. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 12–22, Florence, Italy. Association for Computational Linguistics.
- Anisia Katinskaia, Maria Lebedeva, Jue Hou, and Roman Yangarber. 2022. Semi-automatically annotated learner corpus for Russian. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 832–839, Marseille, France. European Language Resources Association.
- Anisia Katinskaia, Javad Nouri, and Roman Yangarber. 2017. Revita: a system for language learning and supporting endangered languages. In *Proceedings of the joint workshop on NLP for Computer Assisted Language Learning and NLP for Language Acquisition*, pages 27–35, Gothenburg, Sweden. LiU Electronic Press.
- Anisia Katinskaia, Javad Nouri, and Roman Yangarber. 2018. Revita: a language-learning platform at the intersection of ITS and CALL. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Anisia Katinskaia and Roman Yangarber. 2021. Assessing grammatical correctness in language learning. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 135–146.
- Satoru Katsumata and Mamoru Komachi. 2020. Stronger baselines for grammatical error correction using a pretrained encoder-decoder model. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 827–832, Suzhou, China. Association for Computational Linguistics.
- Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. An empirical study of incorporating pseudo data into grammatical error correction. 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 1236–1242, Hong Kong, China. Association for Computational Linguistics.
- Ann Lee, Michael Auli, and Marc’Aurelio Ranzato. 2021. Discriminative reranking for neural machine translation. 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 7250–7264, Online. Association for Computational Linguistics.
- Xia Li and Junyi He. 2021. Data augmentation of incorporating real error patterns and linguistic knowledge for grammatical error correction. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 223–233, Online. Association for Computational Linguistics.
- Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. Corpora generation for grammatical error correction. 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 3291–3301, Minneapolis, Minnesota. Association for Computational Linguistics.
- Zhenghao Liu, Xiaoyuan Yi, Maosong Sun, Liner Yang, and Tat-Seng Chua. 2021. Neural quality estimation with multiple hypotheses for grammatical error correction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for*

- Computational Linguistics: Human Language Technologies*, pages 5441–5452, Online. Association for Computational Linguistics.
- Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. [Encode, tag, realize: High-precision text editing](#). 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 5054–5065, Hong Kong, China. Association for Computational Linguistics.
- Jakub Náplava and Milan Straka. 2019. [Grammatical error correction in low-resource scenarios](#). In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 346–356, Hong Kong, China. Association for Computational Linguistics.
- Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanyskiy. 2020. [GECToR – grammatical error correction: Tag, not rewrite](#). In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online. Association for Computational Linguistics.
- 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.
- Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. [COMET: A neural framework for MT evaluation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.
- Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. [A simple recipe for multilingual grammatical error correction](#). 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 702–707, Online. Association for Computational Linguistics.
- Alla Rozovskaya and Dan Roth. 2019. [Grammar error correction in morphologically rich languages: The case of Russian](#). *Transactions of the Association for Computational Linguistics*, 7:1–17.
- Alla Rozovskaya and Dan Roth. 2021. [How good \(really\) are grammatical error correction systems?](#) In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2686–2698, Online. Association for Computational Linguistics.
- Alexey Sorokin. 2022. [Improved grammatical error correction by ranking elementary edits](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11416–11429, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Felix Stahlberg and Shankar Kumar. 2020. [Seq2Edits: Sequence transduction using span-level edit operations](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5147–5159, Online. Association for Computational Linguistics.
- Felix Stahlberg and Shankar Kumar. 2021. [Synthetic data generation for grammatical error correction with tagged corruption models](#). In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 37–47, Online. Association for Computational Linguistics.
- Craig Stewart, Ricardo Rei, Catarina Farinha, and Alon Lavie. 2020. [COMET - deploying a new state-of-the-art MT evaluation metric in production](#). In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 2: User Track)*, pages 78–109, Virtual. Association for Machine Translation in the Americas.
- Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. [Tense and aspect error correction for ESL learners using global context](#). In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 198–202, Jeju Island, Korea. Association for Computational Linguistics.
- Yujin Takahashi, Masahiro Kaneko, Masato Mita, and Mamoru Komachi. 2022. [ProQE: Proficiency-wise quality estimation dataset for grammatical error correction](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5994–6000, Marseille, France. European Language Resources Association.
- Yujin Takahashi, Satoru Katsumata, and Mamoru Komachi. 2020. [Grammatical error correction using pseudo learner corpus considering learner’s error tendency](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 27–32, Online. Association for Computational Linguistics.
- Maksym Tarnavskyi, Artem Chernodub, and Kostiantyn Omelianchuk. 2022. [Ensembling and knowledge distilling of large sequence taggers for grammatical error correction](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3842–3852, Dublin, Ireland. Association for Computational Linguistics.
- Viet Anh Trinh and Alla Rozovskaya. 2021. [New dataset and strong baselines for the grammatical error correction of Russian](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4103–4111, Online. Association for Computational Linguistics.

- Zhaohong Wan, Xiaojun Wan, and Wenguang Wang. 2020. [Improving grammatical error correction with data augmentation by editing latent representation](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2202–2212, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Max White and Alla Rozovskaya. 2020. [A comparative study of synthetic data generation methods for grammatical error correction](#). In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 198–208, Seattle, WA, USA → Online. Association for Computational Linguistics.
- Zheng Yuan and Ted Briscoe. 2016. [Grammatical error correction using neural machine translation](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386, San Diego, California. Association for Computational Linguistics.
- Zheng Yuan and Christopher Bryant. 2021. [Document-level grammatical error correction](#). In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 75–84, Online. Association for Computational Linguistics.
- Zheng Yuan, Felix Stahlberg, Marek Rei, Bill Byrne, and Helen Yannakoudakis. 2019. [Neural and FST-based approaches to grammatical error correction](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 228–239, Florence, Italy. Association for Computational Linguistics.
- Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. [Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data](#). 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 156–165, Minneapolis, Minnesota. Association for Computational Linguistics.
- Gustavo Zomer and Ana Frankenberg-Garcia. 2021. [Beyond grammatical error correction: Improving L1-influenced research writing in English using pre-trained encoder-decoder models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2534–2540, Punta Cana, Dominican Republic. Association for Computational Linguistics.

## A Cleaning Model’s Output

We have discovered two issues in the hypotheses generated by the GEC model for the validation set. One is extra white spaces in front of hyphenated suffixes added to numbers, e.g., “25 -го апреля” (“*on the 25th of April*”) instead of “25-го апреля”. These extra spaces were removed. Another issue relates to corrections of some short sentences (1-5 words): the generated hypotheses have repeating tokens. It is especially relevant to incomplete sentences in the test set which end with a semicolon, e.g., “Рай :” (“*Heaven :*”). The model is either trying to continue these sentences or just repeating the same word. We have detected all hypotheses for which source sentences were shorter than 6 words (without punctuation) and which include repetitions and replaced them with the source sentences as if they were not corrected by the model at all, in total 44 sentences.

## B Manual Evaluation of Top Hypotheses

We have picked top-3 hypotheses for 100 randomly sampled source sentences from the RULEC test set. These hypotheses were manually evaluated by a native speaker on the following aspect: whether the second-ranked and the third-ranked hypotheses improve the corrections suggested in the top-1 hypothesis or whether the quality of corrections degrades. Manual evaluation has shown that for 58% of checked sentences, the quality only improves with more hypotheses.

## C Paragraph Correction

Due to technical limitations, the GEC model input length cannot exceed 100 tokens. Therefore, to run a preliminary evaluation with whole paragraphs as input, instead of sentences, we had to prune the longest paragraphs to 100 tokens. This leads to losing 107 learner answers. Regarding assessing and detecting grammatical errors, recall drops and precision increases, especially for a setting with re-ranking, see Table ???. As a result, this leads to lower precision for AC answers, since more errors are not corrected. We find several reasons for this decrease in error recall. First of all, the GEC model was pre-trained and fine-tuned on sentences. For example, it corrects erroneous “уже из конца недели” to “уже в конце недели” (“*already at the end of the week*”) only if this error is in a separate sentence. However, the error is not corrected if the model

gets as input a three-sentence paragraph, with this error in the second sentence. Another issue relates to pruning paragraphs which leads to incomplete sentences and broken syntactic relations between words. Paragraph-level assessment requires more research and training GEC models on a wider context, though there are few available datasets for this task.

## D Examples

Table 10 presents more examples where the GEC model generates multiple valid corrections in the same context.

<b>Source:</b> сел рядом на <u>скамеечку</u>
<b>Output 1:</b> сел рядом на <u>скамеечку</u>
<b>Output 2:</b> сел рядом на <u>скамеечке</u>
<b>Output 3:</b> сел рядом на <u>скамейке</u>
<b>Expected:</b> сел рядом на <u>скамеечке</u> ( <i>sat on a bench nearby</i> )
<b>Source:</b> <u>автор</u> работы <u>обнаружил...</u>
<b>Output 1:</b> автор работы <u>обнаружил...</u>
<b>Output 2:</b> <u>авторы</u> работы <u>обнаружили...</u>
<b>Expected:</b> <u>авторы</u> работы <u>обнаружили...</u> ( <i>the authors found...</i> )
<b>Source:</b> Коврин был уже <u>мертвым</u> , когда...
<b>Output 1:</b> Коврин был уже <u>мертв</u> , когда...
<b>Output 2:</b> Коврин был уже <u>мертвым</u> , когда...
<b>Expected:</b> Коврин был уже <u>мертв</u> , когда... ( <i>Kovrin was already dead, when...</i> )
<b>Source:</b> ... <u>хохотал</u> он
<b>Output 1:</b> ... <u>хохотал</u> он
<b>Output 2:</b> ... <u>расхохотался</u> он
<b>Expected:</b> ... <u>хохочет</u> он ( <i>...he laughed</i> )
<b>Source:</b> Большинство заданий <u>выполняется</u> быстро
<b>Output 1:</b> Большинство заданий <u>выполняются</u> быстро
<b>Output 2:</b> Большинство заданий <u>выполняется</u> быстро
<b>Expected:</b> Большинство заданий <u>выполняются</u> быстро ( <i>Most tasks are done fast</i> )
<b>Source:</b> просто <u>удивляюсь</u> и <u>не верю</u> : ты ли это?
<b>Output 1:</b> просто <u>удивляюсь</u> и <u>не верю</u> : ты ли это?
<b>Output 2:</b> просто <u>удивляешься</u> и <u>не веришь</u> : ты ли это?
<b>Expected:</b> просто <u>удивляешься</u> и <u>не веришь</u> : ты ли это? ( <i>just surprised and can't believe, is it you?</i> )
<b>Source:</b> сторожей, подобных этому, я не <u>увидел</u>
<b>Output 1:</b> сторожей, подобных этому, я не <u>увидел</u>
<b>Output 2:</b> сторожей, подобных этому, я не <u>видел</u>
<b>Expected:</b> сторожей, подобных этому, я не <u>увидал</u> ( <i>I have not seen watchmen like this</i> )
<b>Source:</b> проявления мстительности и <u>вредительство</u>
<b>Output 1:</b> проявления мстительности и <u>вредительство</u>
<b>Output 2:</b> проявления мстительности и <u>вредительства</u>
<b>Expected:</b> проявления мстительности и <u>вредительства</u> ( <i>manifestations of revenge and wrecking</i> )
<b>Source:</b> ребенок трогательно <u>погладил</u> моих собак
<b>Output 1:</b> ребенок трогательно <u>погладил</u> моих собак
<b>Output 2:</b> ребенок трогательно <u>поглаживал</u> моих собак
<b>Expected:</b> ребенок трогательно <u>гладил</u> моих собак ( <i>the child touchingly stroked my dogs</i> )
<b>Source:</b> как <u>сформировался</u> этот регион
<b>Output 1:</b> как <u>сформировался</u> этот регион
<b>Output 2:</b> как <u>сформирован</u> этот регион
<b>Expected:</b> как <u>формировался</u> этот регион ( <i>how this region was formed</i> )

Table 10: Examples of some source phrases (“Source”) with learners’ AC answers (blue underlined) which were corrected by the model. “Output 1” and “Output 2” denote the top-2 model’s corrections. “Expected” shows which answers were expected by Revita CALL system.

# “Geen makkie”: Interpretable Classification and Simplification of Dutch Text Complexity

**Eliza Hobo\***  
TNO<sup>+</sup>  
eliza.hobo@tno.nl

**Charlotte Pouw\***  
University of Amsterdam  
c.m.pouw@uva.nl

**Lisa Beinborn**  
Vrije Universiteit Amsterdam  
l.beinborn@vu.nl

## Abstract

An inclusive society needs to facilitate access to information for all of its members, including citizens with low literacy and with non-native language skills. We present an approach to assess Dutch text complexity on the sentence level and conduct an interpretability analysis to explore the link between neural models and linguistic complexity features.<sup>1</sup> Building on these findings, we develop the first contextual lexical simplification model for Dutch and publish a pilot dataset for evaluation. We go beyond previous work which primarily targeted lexical substitution and propose strategies for adjusting the model’s linguistic register to generate simpler candidates. Our results indicate that continual pre-training and multi-task learning with conceptually related tasks are promising directions for ensuring the simplicity of the generated substitutions. Our code repository and the simplification dataset are available on GitHub.<sup>2</sup>

## 1 Introduction

Reading is a foundational skill for acquiring new information. Many sources of information are only available in written form, including educational material, newspaper articles, and letters from municipalities. Although many people learn how to read as a child, not everyone becomes equally skilled at it. In the Netherlands alone, more than 2.5 out of 14 million people over 16 years old are low-literate, meaning that they experience challenges with reading or writing.<sup>3</sup> As a result, they face obstacles in achieving academic success, seeking employment

opportunities, and keeping up-to-date with current events.

One way to address this problem is to reduce text complexity. Texts that contain many infrequent words and complex sentence structures are difficult to read, especially for readers with low literacy and language learners. Automated natural language processing tools for text complexity assessment can help both in assisting editors in the selection of adequate texts and by signaling potential comprehension problems to copywriters. By estimating text complexity, we can select texts that are sufficiently easy for a particular target audience or simplify texts that are too difficult.

Recent neural models for text complexity assessment have obtained good results in classifying texts into discrete categories of complexity (Deutsch et al., 2020; Martinc et al., 2021). The global classification label can be a first indicator but it does not point to specific parts of the input that are complex, leaving it to the human editor to identify the necessary simplifications. In this work, we first explore Dutch complexity prediction on the sentence level (as opposed to full-text classification in previous work) and then zoom in even further.

The complexity of a text is affected by an interplay of various factors, including its structural characteristics, domain, and layout. A crucial component is the choice of the lexical units and their complexity. A system for lexical simplification can support humans in detecting lexical complexity and suggest simpler alternatives. In the sentence *children bear the future, and our resolution to support them determines the world they inherit*, a lexical simplification model could propose to substitute *bear* with simpler words such as *carry*, *hold*, or *shape*. These suggestions can assist human writers in revising and simplifying their text.

Previous approaches to Dutch lexical simplification generated substitution candidates by naively substituting words according to a static alignment

\*Equal contribution.

<sup>+</sup>The experiments were conducted when all authors were affiliated with Vrije Universiteit Amsterdam.

<sup>1</sup>The colloquial Dutch expression “Geen makkie” in the title can be translated as “not easy” or “not a walk in the park”.

<sup>2</sup><https://github.com/clap-lab/makkie/>

<sup>3</sup><https://www.lezenenschrijven.nl/reading-and-writing-foundation>

of synonyms without considering the context of the sentence. This approach does not account for ambiguous words and synonyms that only maintain semantic coherence in a subset of contexts. In the example above, *resolution* can be interpreted as intention, but in the context of TV screens, it refers to sharpness. In order to ensure meaning preservation, lexical simplification needs to be context-sensitive.

**Contributions** We fine-tune BERTje (de Vries et al., 2019), a Dutch pre-trained transformer model, to predict sentence-level complexity and use interpretability methods to show that it captures relevant linguistic cues. We visualize the local attribution values of the model’s predictions in a demo to point end users to complex parts of the sentence. In order to facilitate the simplification process, we introduce LSBertje, the first contextual model for lexical simplification in Dutch. We explore three approaches to adapt the linguistic register of the model, to re-enforce a preference for simplicity in the generated substitutions.

## 2 Related Work

We discuss complexity assessment and lexical simplification as separate consecutive stages in line with related work.

### 2.1 Complexity Assessment

Text complexity is affected by the words we choose and the way we combine them into meaning. The complexity of individual words is determined by features such as length, frequency, morphological complexity, abstractness, and age of acquisition. At the sentence level, syntactic features such as parse tree depth, syntactic ambiguity, and the number of subordinate clauses affect complexity. Features that indicate lexical variety, such as the type-token ratio, can also serve as a proxy for complexity (Schwarm and Ostendorf, 2005; Feng et al., 2009; Vajjala and Meurers, 2012).

Traditional surface-based metrics such as the *Flesch-Kincaid* score are widely used to automatically assess text complexity, but they only consider length characteristics and do not take into account the various intricate factors that influence text complexity. In contrast, feature-based machine learning models leverage numerous features to predict complexity labels, surpassing the capabilities of surface-based metrics (Collins-Thompson and Callan, 2005). Nevertheless, hand-engineering effective features is an expensive and

time-consuming process (Filighera et al., 2019).

Neural models for classifying complexity do not rely on hand-engineered features and show marginal improvements over feature-based models (Deutsch et al., 2020; Martinc et al., 2021), but they lack interpretability. In this study, we analyze if neural models leverage relevant linguistic cues when predicting binary complexity labels for Dutch sentences and can therefore reliably detect sentences that qualify for a simplification procedure.

### 2.2 Lexical Simplification

Lexical simplification characterizes a substitution operation on the lexical level with the goal of reducing the complexity of a sentence and making the text accessible to a wider audience. Lexical simplification of a sentence is typically performed as a pipeline of four consecutive stages: complex word identification, substitution generation, substitution selection and substitution ranking (Sikka and Mago, 2020; Thomas and Anderson, 2012; Paetzold and Specia, 2017b). In this work, we focus on the first two stages.

**Complex Word Identification** In the initial stage, words with simplification potential need to be identified. Traditional approaches for this sub-task use curated lists of complex words (Lee and Yeung, 2018) or word frequency resources to flag words below a certain frequency threshold as complex (Sikka and Mago, 2020). In the most recent shared task for complex word identification (Yimam et al., 2018), feature-based machine learning techniques using length and frequency features obtained the best results. More recent approaches express lexical complexity on a continuous scale (Shardlow et al., 2021) as a binary classification is too simplistic for most educational scenarios. We explore the applicability of gradient-based interpretability techniques for complex word identification (Danilevsky et al., 2020; Sundararajan et al., 2017).

**Substitution Generation** The generation of substitution candidates has traditionally been performed with lexical resources such as WordNet (Miller, 1995; Carroll et al., 1998). In a more data-driven approach, simple-complex word pairs have been extracted from a parallel corpus that aligns sentences in Wikipedia with their counterparts in Simple Wikipedia (Kauchak, 2013; Paetzold and Specia, 2017a). These static approaches are unable



to generate substitution candidates for words that do not occur in the resources or that are spelled differently. In addition, they are prone to generate semantically incoherent candidates since the substitutions are not context-sensitive.

**Context-Aware Substitution Generation** For meaning-preserving simplification, it is important to consider the context of the complex word. Paetzold and Specia (2016b) propose to use the part of speech of a word to narrow down its meaning. Their approach relies on proximity in a static embedding space to find simplifications, which are then disambiguated with respect to their part of speech. As a result, the relatively simple noun *bear* is represented by a different vector than the rather complex verb *bear*. This syntactically informed approach leads to improvements over non-contextualized models, but it still falls short in capturing more fine-grained differences in meaning; even the verb *bear* can be used in a semantic spectrum ranging from *bearing/delivering a child* to *bearing/having a resemblance*.

To capture such subtle distinctions, recent approaches use contextualized language models such as BERT (Devlin et al., 2019) to generate substitutions tailored to the specific context. Alarcón et al. (2021) search the contextual embedding space of a complex word to find context-aware simplification candidates. They find antonyms of the complex word among the generated candidates, which is detrimental to the goal of preserving the meaning of the complex sentence. Qiang et al. (2020) introduce *LSBert*, which uses a prompting strategy based on BERT’s masked language modeling objective to generate context-aware lexical simplification candidates for English sentences. They generate simplifications by masking the complex word. In order to enforce semantic coherence of the masked word, Qiang et al. (2020) feed the input sentences as a duplicated pair and apply the masking operation only on the second sentence. In the recent shared task on multi-lingual lexical simplification (Saggion et al., 2022), approaches that use pre-trained language models produced very competitive results. In all three languages covered in the shared task, English, Spanish, and Portuguese, state-of-the-art results were obtained. In this work, we evaluate the LSBert lexical simplification approach and adapt it to Dutch.

## 2.3 Complexity Assessment and Simplification for Dutch

Work on complexity and simplification for Dutch is sparse. Vandeghinste and Bulte (2019) analyze complexity classification at the document level using feature-based classifiers, but there is currently no known work on neural sentence-level complexity classification for Dutch. Regarding lexical simplification, Bulté et al. (2018) develop a pipeline using various resources. However, systematically evaluating the pipeline is challenging as there is no existing benchmark dataset for lexical simplification in Dutch.

## 3 Complexity Classification

We train a neural classifier for determining binary labels of Dutch sentence complexity and compare its performance to several feature-based classifiers. We then analyze if the neural model captures relevant complexity cues.

### 3.1 Experimental Setup

**Data** We contrast articles from the Dutch newspapers *De Standaard* and *Wablieft* in line with Vandeghinste and Bulte (2019). The two newspapers cover similar topics and events. As *Wablieft* targets an audience that prefers simpler language, the articles are significantly shorter (on average, there are 164 words in *Wablieft* articles vs 383 words in *De Standaard* articles). The source of an article (*Wablieft* vs *De Standaard*) can therefore be easily determined by its length.<sup>4</sup> However, identifying the source is just a proxy for identifying the linguistic characteristics that determine complexity. To go beyond this superficial approach, we instead train our models to predict the complexity of individual sentences.

The corpus contains 12,683 articles from *Wablieft* and 31,140 articles from *De Standaard*.<sup>5</sup> We create a balanced dataset by randomly selecting 12,000 articles from each newspaper and preprocessing them using the same steps as Vandeghinste and Bulte (2019). We split the articles into individual sentences and only keep the first sentence of each article to keep the dataset balanced. We label all sentences from *Wablieft* articles as *easy*

<sup>4</sup>Our BERTje model could distinguish the two types of articles with 99% accuracy when fine-tuned to predict complexity labels for the entire articles.

<sup>5</sup>The data does not include any meta information such as author names and time stamps of publication, which could reveal the source of the article.

and all sentences from De Standaard as *complex*. We use 80% of the data for training, 10% for validation, and 10% for testing. The validation set was used for checking model accuracy at each epoch. Statistics regarding the length and frequency of the words in both types of sentences are shown in Table 1.

	Easy	Complex
#Sentences	12,000	12,000
Word length	<b>4.33</b> (2.14–8.60)	<b>5.10</b> (2.08–11.80)
Word freq.	<b>4.95</b> (1.95–6.38)	<b>4.78</b> (1.39–6.44)

Table 1: Descriptive statistics of the easy and complex sentences that are used to train and evaluate our models. Averages are in bold, ranges are between brackets. Frequencies are measured as standardized Zipf frequencies using the Python package wordfreq.

**Models** We fine-tune a pre-trained transformer model for Dutch sequence classification (BERTje, de Vries et al. (2019)) available from Huggingface and add a linear output layer with ReLU activation and dropout (0.5). The model is optimized using ADAM with a learning rate of 1e-6 and cross-entropy loss.

We use Support Vector Machines (SVM) as our feature-based classification models. We employ the scikit-learn implementation with all default parameters (Pedregosa et al., 2011).

**Complexity Features** Our complexity features can be grouped into three categories: length characteristics, frequency effects, and morpho-syntactic properties. Word frequencies are obtained as standardized Zipf frequencies using the Python package wordfreq (Speer et al., 2018). The package combines several frequency resources, including SUBTLEX lists, e.g. Brysbaert and New (2009), and OpenSubtitles (Lison and Tiedemann, 2016). The morpho-syntactic features are computed using the Profiling-UD tool (Brunato et al., 2020). We calculate all features on the sentence level and train our feature-based models on different combinations of these features. An overview of the features is given in Table 3.

### 3.2 Results

Table 2 shows the prediction accuracy of the fine-tuned BERTje model and several feature-based SVM classifiers for sentence-level complexity classification. We see that the neural model outperforms all feature-based models by 10 percent or

more. For the feature-based classifiers, the best results can be obtained by all types of features (frequency + length + morpho-syntactic), but the morpho-syntactic features only improve the frequency and length-based classifiers with 1 percent accuracy. This might be caused by the fact that the morpho-syntactic features are correlated with length (e.g., parse tree depth naturally increases as the sentence length increases). We conclude that frequency and length are the most predictive features for Dutch sentence-level complexity classification, which is in line with previous work for English (Vajjala Balakrishna, 2015).

Model	Accuracy
Frequency	.72
Frequency + Morpho-Syntactic	.73
Length	.78
Length + Morpho-Syntactic	.79
Frequency + Length	.79
Frequency + Length + Morpho-Syntactic	.80
<b>Neural Model</b> (fine-tuned BERTje)	<b>.90</b>

Table 2: Prediction accuracy of several feature-based SVM models and the fine-tuned BERTje model for sentence-level complexity classification.

**Prediction Confidence** To gain more insight in the linguistic cues that the neural model relies on, we analyze model confidence with respect to the complexity features that our feature-based models were trained on. Table 3 shows the Spearman correlation between complexity features and model confidence for the *complex* class. We see that the model allocates higher probability values to the *complex* class when word length, sentence length, dependency link length, or the number of low-frequency words increases. As the classification is binary, the inverse relationship can be observed for the *easy* class.

Since the correlation values in Table 3 are relatively low, we analyze the corresponding scatter plots. Figure 1 depicts the correlation between model confidence for the *complex* class and the maximum dependency link of the input sentences. We see that low to medium values for the maximum dependency link length do not clearly affect model confidence, but that high dependency link values always lead to high confidence. We observe the same pattern for the other complexity features. This suggests that the model considers relevant complexity features when making its predictions, but that the evidence needs to be strong enough

Category	Linguistic Feature	$\rho$
Length	Avg. word length (# chars)	.41
	Sentence length (# tokens)	.40
Morph-Synt.	Max. dependency link length	.43
	Avg. dependency link length	.40
	# Verbal heads	.37
	Parse tree depth	.35
	Lexical density	.12
Freq	# Low frequency words (Zipf<4)	.37
	Avg word frequency	-.04

Table 3: Spearman correlations between sentence-level complexity features and confidence for the *complex* class of the BERTje model, fine-tuned for sentence-level complexity classification. All positive correlations are significant ( $p < 0.0001$ ). The negative correlation between token frequency and model confidence is not significant ( $p = 0.03$ ).

(i.e., the sentence should be sufficiently complex).

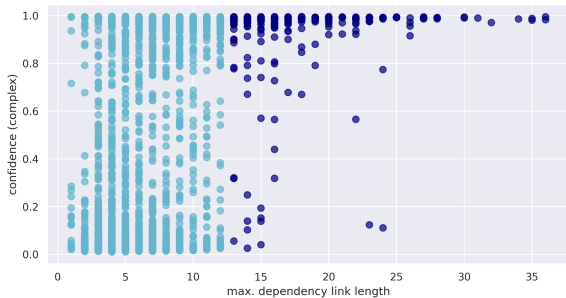


Figure 1: Correlation between BERTje’s confidence for the *complex* class and the maximum dependency link length of the input sentences.

### 3.3 Complex Word Identification

Our results indicate that the fine-tuned BERTje model is a reliable tool for sentence-level complexity classification. It can show an editor which sentences qualify for simplification. Nevertheless, binary complexity classification is an overly simplified operationalization that lacks educational usability. We go one step further and combine the model with feature attribution methods and analyze its utility for the first component of the lexical simplification pipeline: complex word identification.

We implement a demo that explains the predictions of our neural complexity classifier. Users can type Dutch input sentences, which are classified as either *easy* or *complex*. Words that contributed positively or negatively to the model’s prediction are highlighted, as shown in Figure 2. We use Captum (Kokhlikyan et al., 2020) for extracting token-level attributions. Additionally, the sentence-level com-

plexity features from Table 3 are calculated and shown to the user, which give a more fine-grained perspective on the complexity of the input sentence (see Appendix Figure 4).

**Attribution Methods** Selecting the right attribution method is not straightforward. Different attribution methods produce varying, sometimes even contrasting explanations for model predictions (Bastings et al., 2022). Atanasova et al. (2020) find that gradient-based techniques produce the best explanations across different model architectures and text classification tasks. We therefore include three gradient-based attribution methods in our demo: Gradient, InputXGradient, and Integrated Gradients. The vanilla Gradient method estimates feature importance by calculating the gradient (i.e. the rate of change) of a model’s output with respect to a given input feature (Danilevsky et al., 2020). InputXGradient additionally multiplies the gradients with the input, and Integrated Gradients integrates the gradient of the model’s output with respect to the input features along a chosen path between a feature  $x$  and a baseline  $x'$  (Sundararajan et al., 2017). We use the [PAD] token as our baseline.

**Linguistic Plausibility of Attributions** Explanations of the complexity predictions are most useful for end-users of the demo (e.g. teachers) if the attribution scores are linguistically plausible. This means that the scores should match our expectations of what makes a sentence complex or easy to understand. Given the intended use of the demo for complex word identification, we analyze the linguistic plausibility of the attributions with respect to lexical complexity. We expect short and frequent words to receive high attributions when the model predicts that a sentence is easy to understand, while longer and less frequent words should receive high attributions when the model predicts that the sentence is complex.

To better understand the differences between our selected attribution methods and to analyze the linguistic plausibility of the observed patterns, we calculate the Spearman correlation between lexical complexity features and attribution scores. Since our model uses subword tokenization, both attribution scores and complexity features are calculated on the subword level. We exclude the special tokens [CLS] and [SEP] from our analyses.

Table 4 shows that Integrated Gradients is the only method for which the correlations have the ex-

Complexity classification: *complex*

Attribution scores by Integrated Gradients:

De trein -verbinding tussen Gent en Brussel blijft hinder ondervinden

Green words contributed positively to the classification, purple words contributed negatively to it.

Figure 2: Complexity classification and attributions scores for the sentence *De treinverbinding tussen Gent en Brussel blijft hinder ondervinden*, taken from the newspaper De Standaard (translation: the train connection between Ghent and Brussels continues to be affected.) The sentence is classified as complex by the fine-tuned BERTje model. Attributions are calculated by Integrated Gradients.

Class	Method	Len	Freq
Easy	Gradient	.61	-.44
	InputXGradient	.07	.18
	Integrated Gradients	-.10	.19
Complex	Gradient	.54	-.48
	InputXGradient	-.09	.04
	Integrated Gradients	.11	-.14

Table 4: Spearman correlation between subword-level complexity features and subword-level attributions. All correlations are significant ( $p < 0.0001$ .)

pected directionality, i.e. when the model predicts the *easy* class, high attributions are assigned to short/frequent words, and when the model predicts the *complex* class, high attributions are assigned to long/infrequent words. For InputXGradient, we see the opposite pattern, and for Gradient, the directionality of the correlations is the same for both the *easy* and *complex* class. The inconsistency of the three attribution methods is surprising but in line with previous findings (Bastings et al., 2022). More user-centered analyses are required to identify their practical benefits.

To further explore the linguistic plausibility of the attribution scores, we calculate average attribution scores with respect to part-of-speech tags. We again find that the most plausible attributions are generated by the Integrated Gradients approach. In Figure 3, we see that nouns, adverbs, and adjectives are assigned relatively high importance scores when the model predicts the *easy* class. Prepositions, conjunctions, and complementizers receive higher importance when the model predicts the *complex* class. This is plausible since function words often signal a complex sentence structure, while easier sentences typically contain more content words. Additionally, we observe that subwords, which indicate the presence of compound words,

receive higher scores when the model predicts the *complex* class. This is helpful for lexical simplification, as compound words are often challenging to read. Finally, we observe that determiners receive high scores when the model predicts the *easy* class, which aligns with lexical complexity since determiners are short and frequent.

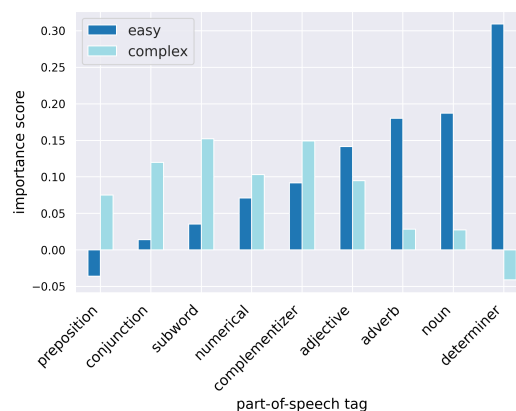


Figure 3: Average attribution scores per part-of-speech tag, generated by Integrated Gradients.

## 4 Context-Aware Simplification

In the second step of the simplification pipeline, we generate context-aware simplifications for Dutch.

**LSBertje** We present *LSBertje*, the first model for contextualized lexical simplification in Dutch. We base LSBertje on LSBert (Qiang et al., 2019, 2020) by altering its language-specific components to Dutch. We replace the language model that generates simplifications with the Dutch BERT model, BERTje. We also replace the stemmer used in filtering with the snowball stemmer.<sup>6</sup>

<sup>6</sup>[nltk.org/api/nltk.stem.snowball.html](https://nltk.org/api/nltk.stem.snowball.html)

## 4.1 Dutch Evaluation Data

Dutch evaluation data for lexical simplification does not yet exist. To evaluate our approach, we develop a pilot benchmark dataset using authentic municipal data. We select sentences from a collection of 15,334 sentences from 48 municipal documents based on the presence of a complex word from a list curated by domain experts and based on their word count (less than 20 words). We exclude incomplete sentences such as headers, sentences without verbs, or with less than four words. From the remaining 6,084 sentences, we randomly sample 250 of complex words from the list and find a sentence for the dataset for 108 of the complex words. Eight sentences where simplification was not possible were removed because: 1) they were part of a named entity, 2) the sentence was incomplete or 3) a simple sense of the word was used. This resulted in 100 sentences.

The sentences were simplified by 23 native speakers of Dutch who pursued or obtained an academic degree. They were shown a sentence with the highlighted complex word and five simplification options that LSBertje generated. The annotators could select from these options and propose additional simplifications. For five sentences, no annotator could come up with a lexical simplification candidate. The remaining 95 sentences contained an average of 2.9 simplification candidates, with a maximum of 7.

## 4.2 Results and Analysis

Table 5 shows that the LSBertje model yields good simplification performance for our dataset. The potential metric shows that the model was able to predict at least one correct simplification candidate in 85% of the sentences. It should be noted that the English benchmark datasets come with a greater variety. In our dataset, a sentence is annotated with 2.9 simplifications on average, whereas BenchLS lists 7.4 substitutions. These size differences can explain the slightly lower potential score and the higher recall for Dutch.

To evaluate the simplicity of the generated substitutions, we assess their frequency using the SUBTLEX-NL corpus (Keuleers et al., 2010) and find that 517 out of 650 generated words occur with higher frequency than the original word. This indicates that the generated simplifications are indeed simpler.

## 5 Register Adaptation Techniques

LSBertje relies on a base model that was pre-trained for masked language modeling and captures aspects of text complexity only as an incidental byproduct. It uses a masked language modeling mechanism that induces semantic preservation by repeating the input sentence. The goal of generating simpler substitutions is only implicitly targeted by restricting the generation to tokens consisting of a single subtoken. This effectively prevents the model from generating infrequent or morphologically more complex words, but the model is not explicitly optimized for capturing different levels of text complexity. We explore three strategies to adapt the linguistic register of the model so that it generates simpler substitutions: conceptual fine-tuning, continual pre-training, and multi-task learning.

**Conceptual Fine-tuning** We aim at adapting the linguistic register of the model by fine-tuning LSBert to predict the linguistic complexity of sentences before applying it for generating substitution candidates. The model is fed a pair of sentences and is trained to predict whether the first sentence is simpler or more complex than the second example. We use sentence pairs from the sentence-aligned simple-complex Wikipedia corpus (Kauchak, 2013). The sentences are balanced with respect to the simplification order condition, and we experiment with the number of sentences.<sup>7</sup>

**Continual Pre-Training** For the second strategy, we adapt the linguistic register by exposing the model to simpler texts using continual pre-training. We continue the pre-training combination of masked language modeling and next-sentence prediction using only sentences from simple Wikipedia.<sup>8</sup> We pair each sentence either with the directly following sentence or with a randomly selected sentence from another Wikipedia article.

**Multi-Task Learning** We then combine the two ideas and train a model on two tasks simultaneously. We use the same training method but replace next-sentence prediction with complexity prediction.

### 5.1 Experimental Setup

As the Dutch dataset is too small for representative evaluation, we first explore the register adaptation

<sup>7</sup>[cs.pomona.edu/~dkauchak/simplification/](http://cs.pomona.edu/~dkauchak/simplification/)

<sup>8</sup>[github.com/LGDoor/Dump-of-Simple-English-Wiki](https://github.com/LGDoor/Dump-of-Simple-English-Wiki)

strategies using English evaluation data and the English LSBert model.

**Evaluation Data** We evaluate the models on three commonly used benchmarking datasets. They consist of sentences from Wikipedia with the complex word highlighted and a list of human-generated simplifications. LexMTurk (Horn et al., 2014), BenchLS (Paetzold and Specia, 2016a) and NNSEval (Paetzold and Specia, 2016b) contain respectively 500, 929, and 239 sentences.

**Implementation Details** We base our implementation on the Huggingface documentation Bert.for.Pretraining and the same model as LSBert.<sup>9</sup> <sup>10</sup> For the masked language modeling components, we mask 15% of the tokens in the input sentences. Optimization is performed using an ADAM optimizer and a batch size of two. The continual pre-training is run for two epochs, the multi-task learning for four epochs. We varied the learning rate (5e-5, 5e-6, 5e-7) and the number of sentences (1000, 10.000, 50.000).

## 5.2 Results

We find that the model adapted with conceptual fine-tuning lost its ability to perform masked language modeling. Its predictions for *bear* in *children bear the future* were: *swallowed, if, knicks, cats, nichol*. These predictions clearly indicate a case of catastrophic forgetting (Liu et al., 2020). In learning a new task, the model forgot its original capabilities.

Both continual pre-training and multi-task learning lead to improved performance on the simplification task in two and three configurations respectively. We find that the configuration of LR 5e-6 and 10.000 sentences is the best for both fine-tuning methods as shown in Table 5. See the Appendix for all scores.

The multi-task learning strategy seems to be the most promising approach. We test the robustness of our findings by training the model using 26 different random seeds. The model outperforms LSBert in 20 cases, see Table 8 of the Appendix for a detailed overview. Overall, we see an increase in precision, recall, and  $F_1$ -score. While the model’s performance is highly sensitive to task-specific components (the learning rate and the num-

ber of sentences), the performance remains robust for variation in the task-independent random seed. The results indicate that multi-task learning is a promising strategy for adapting the model’s linguistic register.

## 5.3 Analysis

We analyze the effect of the register adaptation techniques by comparing the frequency of the generated substitutions using the same resources as Qiang et al. (2019) that contains word frequency counts for Wikipedia articles and a children’s book corpus. We see that the fine-tuned model generates simplifications that occur more frequently compared to the substitutions generated by LSBert (13,030 vs 20,000 occurrences on average). When we zoom in on the generations, we find that the fine-tuned model correctly generates 356 words that were not captured by LSBert and that these words have a high average frequency of 27,000. These findings indicate that the fine-tuning process indeed leads to the generation of simpler words.

## 5.4 Register Adaptation Results for Dutch

Due to the absence of a sentence-aligned simplification corpus for Dutch, we only test the continual pre-training strategy on the Dutch data. The results show that the improvements obtained for English cannot yet be observed for Dutch. In the future, we plan to extend our experiments to a larger dataset and to the multi-task learning strategy.

## 6 Conclusion

In this work, we have introduced two state-of-the-art components for complexity prediction and simplification in Dutch. It can support teachers and text editors in making texts more accessible for people who face reading challenges.

We developed a demo that predicts binary complexity labels for Dutch sentences and highlights words that contributed positively or negatively to the prediction. Additionally, the demo interface provides scales for different aspects of sentence-level complexity to enable a more fine-grained interpretation by the user.

We introduced LSBertje, which is the first model for contextualized lexical simplification in Dutch (to the best of our knowledge). We show that the model can generate adequate simplifications without additional fine-tuning. This base setup can serve as a reasonable starting scenario for context-

<sup>9</sup>bert-large-uncased-whole-word-masking

<sup>10</sup>[https://huggingface.co/transformers/v3.0.2/model\\_doc/bert.html#bertforpretraining](https://huggingface.co/transformers/v3.0.2/model_doc/bert.html#bertforpretraining)

Model	LexMTurk				NNSEval				BenchLS			
	Pot.	P	R	F1	Pot.	P	R	F1	Pot.	P	R	F1
LSBert	98.20	29.58	23.01	25.88	90.79	19.04	25.40	21.77	92.36	23.64	32.08	27.22
Cont. Pre-training	98.40	33.46	26.02	29.28	90.79	20.33	27.14	23.25	92.14	25.68	34.84	29.56
<b>MTL</b>	<b>98.80</b>	<b>33.48</b>	<b>26.04</b>	<b>29.29</b>	<b>92.89</b>	<b>21.55</b>	<b>28.75</b>	<b>24.64</b>	<b>93.54</b>	<b>25.93</b>	<b>35.17</b>	<b>29.85</b>
<b>Dutch Benchmark</b>												
LSBertje	85.26	17.74	65.68	29.16								
Cont. Pre-training	83.16	16.95	59.41	26.37								

Table 5: Simplification performance of the register adaptation techniques as potential (Pot.), precision (P), recall (R), and  $F_1$  for the configuration with a learning rate of 5e-6 and 10,000 fine-tuning sentences.

aware simplification generation for resource-poor languages. We developed a pilot evaluation dataset for Dutch that allowed us to perform initial comparisons. For a more elaborate analysis, a larger Dutch dataset needs to be curated in future work.

We explored strategies to adapt the linguistic register of the model to ensure the simplicity of the generated substitutions and find that both multi-task learning and continual pre-training show considerable potential. We further analyzed the model’s robustness and discovered a strong sensitivity to task-specific hyperparameters but little variation across random seeds.

## Acknowledgements

Eliza Hobo’s simplification experiments were initiated during an internship at the Gemeente of Amsterdam. Iva Gornishka has been a valuable source of insight and support in this process. Charlotte Pouw’s experiments on readability were initiated in a joint project with Florian Kunneman and Bruna Guedes supported by the Network Institute (VU Amsterdam) through the Academy Assistants Program. Lisa Beinborn’s work was supported by the Dutch National Science Organisation (NWO) through the projects CLARIAHPLUS (CP-W6-19-005) and VENI (VI.Veni.211C.039).

## References

Rodrigo Alarcón, Lourdes Moreno, and Paloma Martínez. 2021. Exploration of spanish word embeddings for lexical simplification. In *CTTS@SEPLN*.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. [A diagnostic study of explainability techniques for text classification](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3256–3274, Online. Association for Computational Linguistics.

Jasmijn Bastings, Sebastian Ebert, Polina Zablotskaia, Anders Sandholm, and Katja Filippova. 2022. “will

you find these shortcuts?” a protocol for evaluating the faithfulness of input saliency methods for text classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 976–991, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Dominique Brunato, Andrea Cimino, Felice Dell’Orletta, Giulia Venturi, and Simonetta Montemagni. 2020. [Profiling-UD: a tool for linguistic profiling of texts](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 7145–7151, Marseille, France. European Language Resources Association.

Marc Brysbaert and Boris New. 2009. [Moving beyond Kucera and Francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for American English](#). *Behavior research methods*, 41:977–90.

Bram Bulté, Leen Sevens, and Vincent Vandeghinste. 2018. [Automating lexical simplification in dutch](#). *Computational Linguistics in the Netherlands Journal*, 8:24–48.

John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical simplification of english newspaper text to assist aphasic readers. *Proc. of AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*.

Kevyn Collins-Thompson and Jamie Callan. 2005. [Predicting reading difficulty with statistical language models](#). *Journal of the American Society for Information Science and Technology*, 56(13):1448–1462.

Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. [A survey of the state of explainable AI for natural language processing](#). In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 447–459, Suzhou, China. Association for Computational Linguistics.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Bertjan van Noord, and Malvina Nissim. 2019. [BERTje: A Dutch BERT Model](#). arXiv:1912.09582.

- Tovly Deutsch, Masoud Jasbi, and Stuart Shieber. 2020. [Linguistic features for readability assessment](#). In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–17, Seattle, WA, USA → 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.
- Lijun Feng, Noémie Elhadad, and Matt Huenerfauth. 2009. [Cognitively motivated features for readability assessment](#). In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 229–237, Athens, Greece. Association for Computational Linguistics.
- Anna Filighera, Tim Steuer, and Christoph Rensing. 2019. Automatic text difficulty estimation using embeddings and neural networks. In *Transforming Learning with Meaningful Technologies: 14th European Conference on Technology Enhanced Learning, EC-TEL 2019, Delft, The Netherlands, September 16–19, 2019, Proceedings 14*, pages 335–348. Springer.
- Colby Horn, Cathryn Manduca, and David Kauchak. 2014. [Learning a lexical simplifier using Wikipedia](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 458–463, Baltimore, Maryland. Association for Computational Linguistics.
- David Kauchak. 2013. [Improving text simplification language modeling using unsimplified text data](#). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1537–1546, Sofia, Bulgaria. Association for Computational Linguistics.
- Emmanuel Keuleers, Marc Brysbaert, and Boris New. 2010. Subtlex-nl: A new measure for dutch word frequency based on film subtitles. *Behavior research methods*, 42(3):643–650.
- Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, et al. 2020. Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896*.
- John Lee and Chak Yan Yeung. 2018. [Personalizing lexical simplification](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 224–232, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Pierre Lison and Jörg Tiedemann. 2016. [OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles](#). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).
- Qi Liu, Matt J. Kusner, and Phil Blunsom. 2020. [A survey on contextual embeddings](#). *CoRR*.
- Matej Martinc, Senja Pollak, and Marko Robnik-Šikonja. 2021. [Supervised and unsupervised neural approaches to text readability](#). *Computational Linguistics*, 47(1):141–179.
- George A. Miller. 1995. [Wordnet: A lexical database for english](#). *Commun. ACM*, 38(11):39–41.
- Gustavo Paetzold and Lucia Specia. 2016a. [Benchmarking lexical simplification systems](#). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 3074–3080, Portorož, Slovenia. European Language Resources Association (ELRA).
- Gustavo Paetzold and Lucia Specia. 2017a. [Lexical simplification with neural ranking](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 34–40, Valencia, Spain. Association for Computational Linguistics.
- Gustavo H. Paetzold and Lucia Specia. 2016b. Unsupervised lexical simplification for non-native speakers. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI’16*, page 3761–3767. AAAI Press.
- Gustavo H Paetzold and Lucia Specia. 2017b. A survey on lexical simplification. *Journal of Artificial Intelligence Research*, 60:549–593.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, and Xindong Wu. 2019. Lexical simplification with pre-trained encoders. In *AAAI Conference on Artificial Intelligence*.
- Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, and Xindong Wu. 2020. Lsbert: A simple framework for lexical simplification. *ArXiv*, abs/2006.14939.
- Horacio Saggion, Sanja Štajner, Daniel Ferrés, Kim Cheng Sheang, Matthew Shardlow, Kai North, and Marcos Zampieri. 2022. [Findings of the tsar-2022 shared task on multilingual lexical simplification](#). In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*,



pages 271–283, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Sarah Schwarm and Mari Ostendorf. 2005. [Reading level assessment using support vector machines and statistical language models](#). In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 523–530, Ann Arbor, Michigan. Association for Computational Linguistics.

Matthew Shardlow, Richard Evans, Gustavo Henrique Paetzold, and Marcos Zampieri. 2021. [SemEval-2021 task 1: Lexical complexity prediction](#). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1–16, Online. Association for Computational Linguistics.

Punardeep Sikka and Vijay Mago. 2020. [A survey on text simplification](#).

Robyn Speer, Joshua Chin, Andrew Lin, Sara Jewett, and Lance Nathan. 2018. [Luminosinsight/wordfreq: v2.2](#).

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17*, page 3319–3328. JMLR.org.

S. Rebecca Thomas and Sven Anderson. 2012. [Wordnet-based lexical simplification of a document](#). In *Proceedings of KONVENS 2012*, pages 80–88. ÖGAI. Main track: oral presentations.

Sowmya Vajjala and Detmar Meurers. 2012. [On improving the accuracy of readability classification using insights from second language acquisition](#). In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 163–173, Montréal, Canada. Association for Computational Linguistics.

Sowmya Vajjala Balakrishna. 2015. *Analyzing Text Complexity and Text Simplification: Connecting Linguistics, Processing and Educational Applications*. Ph.D. thesis, Universität Tübingen.

Vincent Vandeghinste and Bram Bulte. 2019. Linguistic proxies of readability: Comparing easy-to-read and regular newspaper dutch. *Computational Linguistics in the Netherlands*, 9:81–100.

Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri. 2018. [A report on the complex word identification shared task 2018](#). In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 66–78, New Orleans, Louisiana. Association for Computational Linguistics.

## A Appendix

LR	No. Sents	Potential	Precision	Recall	$F_1$
NNSEval					
LSBert		90.79	19.04	25.40	21.77
5e-6	1,000	87.03	17.03	22.72	19.47
5e-6	10,000	91.63	<b>20.17</b>	<b>26.91</b>	<b>23.06</b>
5e-6	50,000	90.79	<b>20.33</b>	<b>27.14</b>	<b>23.25</b>
5e-7	1,000	88.70	17.95	23.95	20.52
5e-7	10,000	88.28	17.24	23.00	19.71
5e-7	50,000	88.28	17.53	23.39	20.04
LexMTurk					
LSBert		98.20	29.58	23.01	25.88
5e-6	1,000	95.80	25.64	19.94	22.43
5e-6	10,000	<b>98.60</b>	<b>32.16</b>	<b>25.01</b>	<b>28.14</b>
5e-6	50,000	<b>98.40</b>	<b>33.46</b>	<b>26.02</b>	<b>29.28</b>
5e-7	1,000	97.20	26.76	20.81	23.41
5e-7	10,000	96.00	25.89	20.13	22.65
5e-7	50,000	97.80	26.62	2070	23.29
BenchLS					
LSBert		92.36	23.64	32.08	27.22
5e-6	1000	88.37	20.13	27.32	23.18
5e-6	10,000	<b>92.68</b>	<b>24.74</b>	<b>33.57</b>	<b>28.48</b>
5e-6	50,000	92.14	<b>25.68</b>	<b>34.84</b>	<b>29.56</b>
5e-7	1,000	90.42	21.33	28.95	24.57
5e-7	10,000	89.34	20.56	27.90	23.68
5e-7	50,000	91.50	21.42	29.07	24.67

Table 6: Performance of the Continual Pre-training Setup on the Benchmarking Datasets for Different Experimental Conditions

LR	Num Sents	Potential	Precision	Recall	$F_1$
BenchLS					
	LSBert	92.36	23.64	32.08	27.22
5e-5	1,000	87.19	21.77	29.54	25.06
5e-5	10,000	91.17	<b>24.92</b>	<b>33.82</b>	<b>28.69</b>
5e-6	1,000	89.99	21.07	28.59	24.26
<b>5e-6</b>	<b>10,000</b>	<b>93.54</b>	<b>25.93</b>	<b>35.17</b>	<b>29.85</b>
5e-6	50,000	92.03	<b>24.19</b>	<b>32.82</b>	<b>27.85</b>
5e-7	1,000	84.61	18.62	25.26	21.43
5e-7	10,000	88.91	20.23	27.45	23.29
5e-7	50,000	90.74	22.37	30.35	25.76
LexMTurk					
	LSBert	98.20	29.58	23.01	25.88
5e-5	1,000	97.00	28.54	22.20	24.97
5e-5	10,000	98.00	<b>32.22</b>	<b>25.06</b>	<b>28.19</b>
5e-6	1,000	96.60	26.76	20.81	23.41
<b>5e-6</b>	<b>10,000</b>	<b>98.80</b>	<b>33.48</b>	<b>26.04</b>	<b>29.29</b>
5e-6	50,000	98.20	<b>30.92</b>	<b>24.05</b>	<b>27.05</b>
5e-7	1,000	94.20	24.55	19.09	21.48
5e-7	10,000	96.00	25.98	20.21	22.73
5e-7	50,000	97.00	28.16	21.90	24.64
NNSEval					
	LSBert	90.79	19.04	25.40	21.77
5e-5	1,000	84.52	17.07	22.78	19.52
5e-5	10,000	<b>91.21</b>	<b>19.29</b>	<b>25.74</b>	<b>22.05</b>
5e-6	1,000	87.45	18.20	24.29	20.81
<b>5e-6</b>	<b>10,000</b>	<b>92.89</b>	<b>21.55</b>	<b>28.75</b>	<b>24.64</b>
5e-6	50,000	<b>92.05</b>	<b>19.71</b>	<b>26.30</b>	<b>22.53</b>
5e-7	1,000	81.59	14.81	19.77	16.93
5e-7	10,000	86.61	17.36	23.17	19.85
5e-7	50,000	87.45	18.74	25.01	21.43

Table 7: Performance of the Multi-Task Learning Setup on the Benchmarking Datasets for Different Experimental Conditions

Random Seed	Potential	Precision	Recall	<i>F1</i>
1	84.94	18.12	24.18	2.71
2	88.28	18.08	24.12	20.66
<b>3</b>	<b>92.89</b>	<b>21.55</b>	<b>28.75</b>	<b>24.64</b>
4	88.28	19.29	25.74	22.05
5	90.79	20.33	27.14	23.25
6	86.61	17.62	23.51	20.14
7	91.63	19.87	26.52	22.72
8	90.79	20.75	27.69	23.73
9	87.03	19.08	25.46	21.81
10	88.28	19.67	26.24	22.48
11	90.79	20.17	26.91	23.06
12	89.54	20.59	27.47	23.54
13	92.89	21.00	28.03	24.01
14	90.79	20.96	27.97	23.97
15	85.77	18.45	24.62	21.10
16	88.70	18.70	24.96	21.38
17	89.54	18.83	25.13	21.53
18	91.63	20.67	27.58	23.63
19	89.54	20.00	26.69	22.87
20	88.28	16.86	22.50	19.28
21	90.38	19.21	25.63	21.96
22	88.28	17.91	23.90	20.47
23	92.05	20.88	27.86	23.87
24	87.45	18.20	24.29	20.81
25	93.31	20.92	27.92	23.92
26	90.79	19.96	26.63	22.82
mean	89.59	19.53	26.06	22.32

Table 8: Multi-task learning results for NNSEval with varying random seeds. The learning rate is fixed at  $5e-6$  and fine-tuning is conducted on 10,000 sentences.

DIFFICULTY OF LINGUISTIC FEATURES

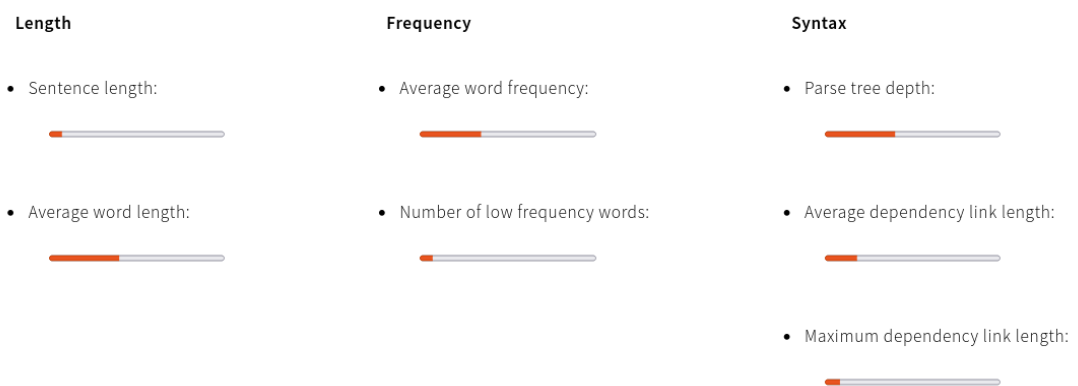


Figure 4: Complexity features for the sentence *De treinverbinding tussen Gent en Brussel blijft hinder ondervinden*, taken from the newspaper De Standaard (translation: the train connection between Ghent and Brussels continues to be affected.) The sentence is classified as complex.

# CEFR-based Contextual Lexical Complexity Classifier in English and French

Desislava Aleksandrova and Vincent Pouliot

CBC/Radio-Canada

dessy.aleksandrova@radio-canada.ca, vincent.pouliot@radio-canada.ca

## Abstract

This paper describes a CEFR-based classifier of single-word and multi-word lexical complexity in context from a second language learner perspective in English and in French, developed as an analytical tool for the pedagogical team of the language learning application *Mauril*. We provide an overview of the required corpora and the way we transformed it into rich contextual representations that allow the disambiguation and accurate labelling in context of polysemous occurrences of a given lexical item. We report evaluation results for all models, including two multi-lingual lexical classifiers evaluated on novel French datasets created for this experiment. Finally, we share the perspective of *Mauril*'s pedagogical team on the limitations of such systems.

## 1 Introduction

The lexical complexity classification task exists in its simplest form as the binary complex word identification task (CWI) and at its most complex, as a multi-class classification where the class nomenclature and granularity is determined by the labelling of the training data. Lexical complexity finds application in text and lexical simplification systems, in automated language proficiency assessment, as well as in the creation of level-appropriate pedagogical content, which also happens to be the use case of *Mauril*<sup>1</sup>.

*Mauril* is a new, free digital platform leveraging a wide range of stimulating and entertaining content from CBC and Radio-Canada to help users learn English and French. Financed and endorsed by the Government of Canada, this new tool is designed and deployed by CBC/Radio-Canada, in collaboration with a committee of pedagogical experts. It's meant to help improve oral comprehension and integrate language knowledge in everyday life.

<sup>1</sup><https://mauril.ca/en/>

The language learning process in *Mauril* begins with a placement test and is then organized by levels covering beginner, intermediate and advanced proficiency<sup>2</sup>. Each level contains units and each unit consists of a video clip of varying length (anywhere from 1 to 22 min) punctuated by comprehension questions and accompanied by highlighted vocabulary (words and expressions) with a corresponding difficulty level.

The creation of pedagogical content (from the selection of video segments, through questions and vocabulary definition to the assignment of difficulty level) is performed manually by experienced foreign language teachers. This labour intensive process of content creation was the target of a lexical processing pipeline designed to streamline and facilitate the extraction and addition of more level-appropriate vocabulary to all existing units. The system in question had to be able to parse the subtitle file of a video segment, reconstruct and then segment the text into tokens, detect and extract multi-word expressions and then assign a complexity label to all occurrences of words and expressions in context. The central component of this system and the current publication is a CEFR-based<sup>3</sup> lexical complexity classifier for both French and English.

In this paper, we apply a novel approach to lexical complexity prediction (LCP), based on rich contextual representations. We show that our system is capable of:

- classifying word senses in context;
- predicting complexity of both words and phrases;
- producing results in French with no or limited training data

<sup>2</sup>cf. § 5.1 for a mapping between *Mauril*'s levels and other standards for language ability assessment.

<sup>3</sup>The *Common European Framework of Reference (CEFR)* is a common basis for the elaboration of pedagogical materials and an international standard for describing the proficiency of foreign-language learners (Council of Europe, 2001).

## 2 Related work

In the context of their language-learning platform offering a digital language proficiency assessment exam, DuoLingo had developed and released a CEFR checker (now discontinued) allowing users to validate the difficulty of words and text in English and Spanish. The lexical complexity component of the tool was described in [Settles et al. \(2020\)](#) as a CEFR-based vocabulary scale model based on CEFR vocabulary wordlist (an inventory of 6,823 English words labelled by CEFR level, mostly in the B1/B2 range). The authors proposed two regression models fit on lexical item representations composed of surface-based features aimed as a proxy of frequency. The models did not seem to handle multi-word expressions, nor common contractions such as *doesn't* and *you've*. Their complexity predictions were lemma-based and did not take inflection into account, which was evident and consequential in Spanish more than it was in English. Finally, the misclassifications reportedly attributed to polysemy ([Settles et al., 2020](#), p.6) were in fact cases of homonymy, since the representations did not include PoS information.

Disambiguating polysems is a challenge for all lemma-based complexity corpora ([FLE, 2004](#); [Lété, 2004](#); [Cobb, 2007](#); [Lonsdale and Le Bras, 2009](#); [François et al., 2014](#); [Schmitt et al., 2021](#)) which conflate polysemous entries into a single entry and assign it a single level. However, not all senses of a polysemous word are learned at once and the different meanings of polysemous words are not uniformly distributed across texts of varying difficulty. [François and Watrin \(2011\)](#) even found a negative association between frequency and complexity with more frequent words being associated with more complex texts. This may be attributable to the fact that frequent words tend to be more polysemous ([Zipf, 1945](#)) and complex texts are likely using more than one of those meanings disguised as occurrences of the same lemma. In fact, learners encounter highly polysemous words most often ([Crossley et al., 2010](#)), hence the importance of disambiguating and accurately predicting the complexity of word senses.

The role of context in LCP is two-fold. Firstly, it is crucial in deriving the correct sense of a polysemous word (word sense disambiguation), as words in isolation provide no information as to their intended meaning. Secondly, it has an incidence on a word's complexity as a source of complementary

information. Learners acquire much of their vocabulary knowledge from context rather than from decontextualized forms such as word lists, definitions, etc.) ([Nagy, 1995](#)) [Gooding and Kochmar \(2019a\)](#) were some of the first to recognize the importance of context for the task of CWI. As they correctly pointed out, the perceived complexity of the lexeme *molars* in the phrase *Elephants have four molars...* may be higher than in the phrase *... new molars emerge in the back of the mouth.* since the second occurrence is surrounded by familiar words that imply its meaning, while the first co-occurs with the rarer and less semantically similar *elephants*.

In more recent work, ([Alfter and Volodina, 2018](#); [Alfter, 2021](#)) found that one of the most important predictors of complexity in their experiments was topic distribution – a context feature modelling polysemy and defined as a vector indicating all topics under which a word occurred. Effects of the inclusion of context on predicting lexical complexity are also discussed in a recent survey of LCP by ([North et al., 2023](#)).

In contrast, lexical complexity work on French has mostly focused on representing and classifying lexical items in isolation ([Gala et al., 2013](#); [François et al., 2016](#)), independently of the context in which they appear. This position is reflected in the lexical complexity corpora available in French ([François et al., 2014](#); [Lété, 2004](#)) which provides no means of contextualizing or disambiguating word senses. [Gala et al. \(2014\)](#) have presented lexical classification models trained on these corpora where lexical items were represented by 49 orthographic, morphologic and statistical features. Their L2 classifier achieved 43% accuracy on the six-way classification task.

Approaches based on such linguistic features often struggle to represent MWEs since the latter are absent from vocabulary lists despite their high frequency in everyday interactions<sup>4</sup> and invite the use of simplifying techniques such as averaging the constituents of the MWE (which in turn wrongly assume compositionality). At the same time, studies in both French and English have shown the importance of MWE-based features for the accurate assessment of lexical complexity ([François and Watrin, 2011](#); [Kochmar et al., 2020](#)).

---

<sup>4</sup>[Jackendoff \(1995\)](#) estimates that not less than half of the lexical units readily available to a speaker in daily interactions are MWEs.

### 3 Training data

To train a contextual classifier of lexical items, we needed a collection of words and expressions associated with complexity levels and accompanied by at least one sentence illustrating their usage in context.

#### 3.1 English

For the English classifier, we used the Cambridge University Press’s English Vocabulary Profile <sup>5</sup> (Capel, 2010, 2012), following Settles et al. (2020). The EVP corpus is a rich resource in British and American English which associates single words, phrasal verbs, phrases, and idioms (Table 1) not only with a CEFR level and a part of speech tag (PoS), but with a definition, a dictionary example and production examples on the basis of several hundred thousand examination scripts written by learners from all over the world. It offers reliable information about which words (and more importantly, which meanings of those words) ARE known and used by learners (rather than SHOULD be known) at each level of the CEFR. For example, we find 10 entries for the word form *run* in the American English section of the corpus, two noun forms and eight verbs, whose complexity varies between A1 (*He can run very fast.*) and C2 (*He would like to run for mayor.*) Each of those meanings is accompanied by usage examples taken from essays of students whose acquisition level corresponds to the complexity level of the word. Such contextual examples allowed us to include disambiguated polysemous lexemes with varying complexity to the training data.

Word form	POS	Level
sleep	verb	A1
sleep with sb	phrasal verb	C2
lose sleep over sth	idiom	C2
not sleep a wink	phrase	C2

Table 1: Example entries from the EVP corpus

After extracting all triplets <word form, level, examples> from the American subset of the corpus, we made sure that each word form’s inflection matches the inflection of its occurrence in at least one usage example. Those who differed were modified manually to assure such correspondence. Uninflected phrases such as *not sleep a wink* be-

<sup>5</sup><https://www.englishprofile.org/wordlists/evp>

came *didn’t sleep a wink* to include the auxiliary verb present in the usage example. Phrasal verbs and expressions with placeholder arguments such as *sleep with sb*, *rush into sth* lost the arguments. Placeholder arguments in non-contiguous expressions such as *grab sb’s attention* were replaced by actual arguments from the entry’s usage examples: *grab the reader’s attention*, *grab people’s attention*. Complex items with word order variation such as *set back sb/sth* or *set sb/sth back* were split into multiple word forms. Following the edits, the dataset contained 14,177 entries distributed unevenly in six classes (Table 2) of which 90% were used for training and the remaining 10% were kept for evaluation.

A1	A2	B1	B2	C1	C2
804	1525	2715	3829	2159	3145

Table 2: Class distribution of the EVP corpus

#### 3.2 French

To our knowledge, no lexical complexity corpora in French resembles *EVP* and its disambiguated, contextualized, CEFR labelled words and expressions extracted from ESL production corpora.

*FLELex* (François et al., 2014) is a graded lexicon for French as a foreign language (FFL) that reports the normalized frequencies of words (lemmas) across CEFR levels. The frequency distributions have been estimated on a corpus of FFL textbooks and FFL simplified books rather than on learners’ corpora. Polysemous lemmas in *FLELex* are ambiguous and conflate the frequencies of all lexemes with the same spelling. As a consequence, the associated frequency distribution is likely right-skewed, reflecting the relatively higher frequencies of easier meanings. In addition to lacking usage examples, it requires a mapping between frequency distributions and CEFR classes (Gala et al., 2013; Alfter et al., 2016; Pintard and François, 2020).

*Manulex* (Lété, 2004) offers frequency distributions of 23K+ French lemmas and 43K+ word forms across three primary school levels rather than CEFR. As a further limitation, the corpus contains no usage examples.

*A Frequency Dictionary of French: Core Vocabulary for Learners* (Lonsdale and Le Bras, 2009) enumerates the 5000 most frequent lemmas with a usage example in French and absolute frequency among other attributes. Word frequency, how-



ever, is a necessary but not sufficient predictor of lexical complexity. *LexTutor*'s frequency lists (Cobb, 2007) contain only lemmas and no complexity labels or usage examples. *Les référentiels* (FLE, 2004) compile lexical inventories across most CEFR levels based on target competence rather than on actual learner performance. Very few lemmas are paired with a sentence, but an organization by themes makes it possible to manually disambiguate homographs and polysemous words within and across complexity levels.

## 4 Evaluation data

### 4.1 English

To evaluate the classifiers we trained on English data, we used 10% of the EVP corpus. This approach to evaluation, despite being methodologically sound, has a tendency to overestimate performance since evaluation and training data have the same distribution.

### 4.2 French

To evaluate the models' performance on French, we had to create labelled data since none was readily available. The current section describes three versions of our French Evaluation Corpus (FEC) two of which are based on parts of *Les référentiels* (FLE, 2004), a series of word indexes that serves as a lexical reference for learners from levels A1 to C1. Each level is subdivided into chapters that, in turn, break the lists of words and expressions into different themes. Some lexical items are accompanied by context sentences.

We first transcribed 13,016 words and expressions (for levels A1 to B2) with their corresponding PoS and examples, whenever available. Since the advanced level C1 was out of scope for Mauril's use case, we only extracted 10 examples from it. We then excluded all vocabulary belonging to European varieties of French (e.g. *atriaux*, *boutefas*, *longeole*, *schublig*, all types of sausages from Switzerland) and only kept lexical units actively used in Québec. We also identified and erased many duplicate entries through and across levels, while manually disambiguating and preserving occurrences of polysems. The last systematic edit we made to the list was to omit MWEs that were considered non-productive or redundant.

For the first version of the corpus (FEC1), we kept only entries which already had context examples for a total of 914 (Table 3). Despite the

extensive cleaning and editing, we found that the corpus still contained many inconsistencies which motivated the creation of other versions.

contre	B1	Mets cette chaise <b>contre</b> le mur.
against		Put this chair <b>against</b> the wall.
contre	B1	Je suis <b>contre</b> son projet.
against		I am <b>against</b> his project.

Table 3: Examples from the FEC1 corpus

For the second version (FEC2), we extracted the lexical items from several themes (semantic fields) across all levels, making sure to avoid the contradictions present in the first version by not allowing multiple occurrences of the same lexeme (at any level). The resulting list contained 473 lexical items (A1: 83, A2: 99, B1: 114, B2: 167, C: 10) most of which did not have a corresponding example in *The Référentiels*. We had usage examples created for all lexemes by a trained linguist, native speaker of French. Since sentences were aimed to be understandable in isolation, most of them followed a simple, declarative SVO structure with very few having subordinate clauses, complex noun phrases or non-pronominal subjects. Still, we were unable to make sure that the sentence complexity of each example is equal or lower to the complexity level of the lexical item whose usage it aimed to illustrate (the way a performance corpus such as *EVP* naturally does).

The third version of the corpus (FEC3) is based on a series of FSL<sup>6</sup> textbooks from Quebec (Gouvernement du Québec, 2014) covering levels A1 to B2 and targeting adult learners (Table 4). By extracting vocabulary (in context) from listening and reading comprehension activities, we could better control for the difficulty of the usage examples, even though the resulting complexity labels still equate comprehension rather than production. FEC3 is the smallest and most *Quebécois* corpus of the three with 48 lexical items in each of the 4 levels. While compiling the corpus, we noticed that it contained more advanced words taught at lower levels than the previous source. We attribute it to the didactic materials being developed following the FLI<sup>7</sup> approach and targeting adults integrating a new country.

All three versions of the FEC reflect competence rather than performance, contrary to the training

<sup>6</sup>French as a second language

<sup>7</sup>French Language of Integration

imbibez	A2	<b>Imbibez</b> un linge de vinaigre chaud ou froid.
soak		<i>Soak</i> a cloth in hot or cold vinegar.
compte	A2	Si vous disposez de fonds dans votre <b>compte</b> , vous pouvez envoyer de l'argent dans le monde entier.
account		If you have funds in your <b>account</b> , you can send money worldwide.

Table 4: Examples from the FEC3 corpus

data used to create the classifier.

## 5 Lexical classification

In this section, we describe the creation of a classifier able to assign a complexity level between  $1 \equiv A1$  and  $6 \equiv C2$  to the meaning of any word or multi-word expression as determined by its context.

### 5.1 Classes

Lexical complexity may be cast as a 6-class classification problem whenever training data is available for all CEFR levels. Mauril’s pedagogical content is distributed among eight levels and covers two of the three proficiency stages defined in the Canadian Language Benchmark’s nomenclature: Basic and Intermediate<sup>8</sup>. These eight levels correspond to four of the CEFR levels, as illustrated in Figure 1. Given the relatively small number of examples in A1 and A2 (cf. Table 2) and Mauril’s coverage, the lexical classification need of Mauril is better satisfied by a 4-class classifier with a combined class for both the beginner and the advanced levels. In this way, each of the beginner, intermediate and advanced levels in Mauril corresponds to a class label with the fourth label C covering advanced vocabulary beyond the current pedagogical scope of the application (Figure 1).

### 5.2 Preprocessing

The minimal preprocessing of the triplets targets the word form and the examples and consists of tokenization using spaCy’s models for English and French<sup>9</sup>. An additional preprocessing step of expanding some common unambiguous contractions

<sup>8</sup>The two Advanced levels in Mauril correspond to CLB’s levels 7 and 8, both belonging to the Intermediate proficiency stage

<sup>9</sup><https://spacy.io/ | v. 3.1.3 | en-core-web-lg, fr-core-news-lg>

CLB & MAURIL	CEFR & EVP	EVP 4
Beginner 1	A1	A
Beginner 2		
Beginner 3	A2	
Beginner 4		
Intermediate 1	B1	B1
Intermediate 2		
Advanced 1	B2	B2
Advanced 2		
	C1	C
	C2	

Figure 1: Class mappings between CLB & Mauril levels, the six levels of CEFR & EVP, and the rebinned version of the EVP corpus with four classes

in English (e.g. *don’t* → *do not*) improves the tokenization.

### 5.3 Vectorization

Rather than representing the vocabulary items by their frequency and/or surface-level characteristics (e.g. number of characters, number of syllables, etc.), we obtain a semantic, contextual, dense vector representation of each item from a pre-trained masked language model (Devlin et al., 2018).

Unlike word2vec models which are sources of non-contextualized (or static) embeddings, trained masked language models such as BERT assign a different representation to each instance of a word in a different context. Garí Soler and Marianna Apidianaki (2021) showed that nonetheless, such language models encode information about a word’s monosemous or polysemous nature. Their experiments also showed that the uncased BERT model possessed more knowledge about lexical polysemy than the cased one.

To obtain a vector representation reflecting a particular meaning of a string, we encode (using the model bert-base-uncased<sup>10</sup>) each of the usage examples of a triplet containing the string and then select the WordPieces<sup>11</sup> composing it. For each WordPiece, we extract and sum the vector representations from the 12 hidden layers. Finally, we aggregate the vectors of all WordPieces by averaging them. When more than one usage examples

<sup>10</sup><https://huggingface.co/bert-base-uncased>

<sup>11</sup>Subwords resulting from a segmentation algorithm

are accompanying a word form, we take the mean of all occurrences as a final representation. We considered different selection and pooling strategies for the hidden layer representations: first, last, second-to-last layers, summing or concatenating the last four hidden layers. The SVC model trained on the sum of all 12 hidden layers achieved the highest accuracy in a 3-fold cross validation.

In this manner, embeddings of tokens with the same or with similar meanings are more alike (in terms of cosine similarity) despite the varying context, than embeddings of homographs with unrelated senses.

Table 5 illustrates eight occurrences of the token *run* in contextual minimal pairs, where each context evokes a different meaning (present in the EVP dataset). We compared the embeddings of the same token in each of the new contexts in Table 6 to the vectors in Table 5 to find the closest meaning in terms of pairwise cosine similarity. The experiment shows that as long as their contexts evoke the same meaning, the embeddings of two occurrences of the same word would remain very similar.

#	WORD FORM IN CONTEXT	MEANING
1	I <b>ran</b> a marathon	MOVE FAST
2	I <b>ran</b> the program	OPERATE
3	I <b>ran</b> into trouble	ENCOUNTER
4	I <b>ran</b> into the kitchen	ENTER
5	I <b>ran</b> into a friend	MEET
6	I <b>ran</b> an ad	PUBLISH
7	I <b>ran</b> the water for 20 min	LIQUID
8	I <b>ran</b> for president	ELECTION

Table 5: Minimal pairs of sentences illustrating different meanings of the word form *ran*

## 5.4 Classification

We used a support vector classifier algorithm<sup>12</sup> (Platt et al., 1999) with adjusted class weights inversely proportional to class frequencies in the input data to correct for the class imbalances. For the same reason, we calculate and report a balanced accuracy score<sup>13</sup> defined as the macro-average of recall scores per class.

## 5.5 Transfer learning

In the absence of appropriate training data in French, we used a transfer learning approach

<sup>12</sup>[sklearn.svm.SVC](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html)

<sup>13</sup>[sklearn.metrics.balanced\\_accuracy\\_score](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.balanced_accuracy_score.html)

WORD FORM IN CONTEXT	CLOSEST MEANING	COS. SIM.
He <b>ran</b> 24 miles	MOVE FAST	0.89
She <b>ran</b> the race	MOVE FAST	0.87
You <b>ran</b> the script	OPERATE	0.90
He <b>ran</b> into problems	ENCOUNTER	0.94
The car <b>ran</b> into a pothole	ENCOUNTER	0.88
She <b>ran</b> for mayor	ELECTION	0.92
He <b>ran</b> for office	ELECTION	0.92
I <b>ran</b> into the house	ENTER	0.99
He <b>ran</b> into the president	MEET	0.94
I <b>ran</b> into you	MEET	0.95

Table 6: The word form *ran* in different contexts with its corresponding closest meaning in terms of cosine similarity

to train a multilingual lexical classifier. We replaced the monolingual source of embeddings with `bert-base-multilingual-uncased`<sup>14</sup> and trained a classifier on the new representations of the English training data. The resulting model is capable of encoding and classifying multilingual input, including in French by leveraging correlations present in the monolingual training data. We hypothesize that even though the lexicalization of senses and their associated complexity varies across languages, there are reliable regularities between form, meaning, and difficulty present in many languages, especially closely related ones (such as English and French). The accuracy of feature-based lexical classifiers has showed that between 40 and 65% of the variance in lexical complexity models can be explained by universal properties such as frequency, word length and other stylistic characteristics (Gala et al., 2014; Alfter and Volodina, 2018; Alarcon et al., 2019). Ideally, the approach of transfer learning should be applied from morphologically richer languages (such as French) to languages with less inflectional variability (such as English), provided training data is available.

## 6 Results and Discussion

To establish the effectiveness of feature-based representation as lexical complexity predictors on the EVP dataset, we trained the model ME6 Baseline, a support vector classifier (with `class_weight="balanced"`) fitted on frequency and two common surface features: the length of

<sup>14</sup><https://huggingface.co/bert-base-multilingual-uncased>

the word form in characters and in tokens. Without contextual information, we could only disambiguate some of the homographs by part-of speech and had to reduce multiple occurrences of a single token+POS pair to the one with the lowest complexity level. This resulted in 11,133 data points of which we used 90% for training and 10% for evaluation. The resulting confusion matrix with normalized scores per class on Figure 2 shows poor recall for all inner classes, especially the A2 level.

We then trained a model called ME6 Contextual which fits the same support vector classifier on the 6-class training set of the EVP corpus (cf. §3.1) with word embeddings extracted from a monolingual language model (as described in §5.3). For this experiment, we could disambiguate and use all training points, including polysems, resulting in a larger training set (12,760). The evaluation on 10% of the corpus (1,418 data points) produced the results on Figure 3. The improved performance of the contextual model is consistent across all six classes and visible on both the confusion matrix as precision and recall and Table 7 in terms of F1 scores. Classification errors are limited to the neighbouring classes.

We further trained a 4-class classifier (ME4 Contextual) on a rebinned and rebalanced<sup>15</sup> version of the dataset. The reduced number of classes provides a further improvement of F1 scores (Table 7) despite the reduction in training data caused by the rebalancing.

To train the multilingual model MME4 Contextual, capable of classifying not only English but also French words and expressions, we used the method described in section 5.5. When evaluated on English data, the model performs almost as well as the one trained on contextual vectors from a monolingual BERT (Table 7). When evaluated on French data however (cf. § 4.2), there seems to be a significant drop in performance, most noticeable in the intermediate classes. The model overestimates the complexity of all classes by predicting the label C for the majority of examples which explains the poor F1 scores of the C class.

The last model we evaluated, MMEFR4 Contextual, was trained on a combination of English and French labelled data from EVP and FEC1. Despite the shortcomings of the evaluation

corpora we produced in French, it could be used for training, especially the FEC1 which has 904 examples labelled from A1 to B2<sup>16</sup>. After rebalancing the resulting joined dataset, we trained the same support vector classifier on 99% of the data, leaving 1% for evaluation.

The results listed in Table 7 show a significant improvement of the F1 scores for all classes (except for C where the complexity of the 10 examples in the evaluation sets is now underestimated) on FEC2 and FEC3. The scores on English data confirm that the gain in French was not achieved at the expense of the performance in English.

We will be releasing<sup>17</sup> code and English data used for training as well as trained models with the exception of any model trained on a combination of English and French data.

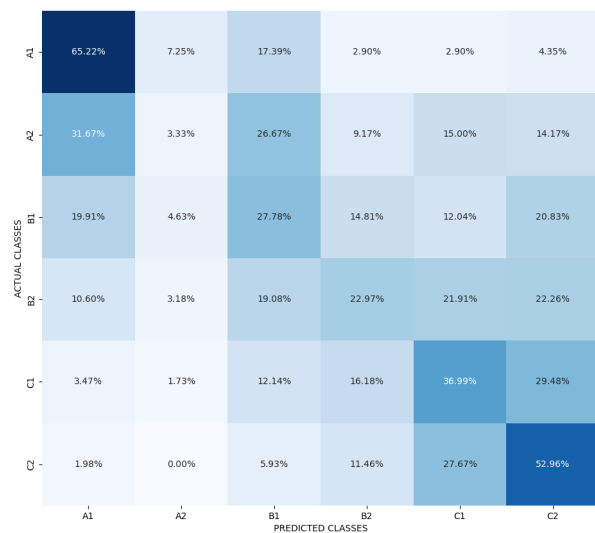


Figure 2: Confusion matrix with normalized scores per class of model ME6 Baseline

Analysis of the errors on the French evaluation corpora showed the significance of the context’s complexity for the individual lexical item’s complexity prediction. Naturally, contextual representations of lexical items are influenced by the surrounding words and syntactic structures, but the extent to which this affects the lexical classifier becomes more visible in a competence type of corpora such as FEC1-3. Furthermore, analysis of the errors on French corpora show that when the model has only seen English data, it has a tendency to overestimate the complexity of inflected French verbs since the training data does not reflect the

<sup>15</sup>To balance the classes, we reduced the size of the largest class C to the size of the second largest – B2.

<sup>16</sup>We excluded the ten examples of the C class since those are present in FEC2 and FEC3

<sup>17</sup><https://github.com/cbrc/vocabclf>

Model	Lang.	A1	A2	B1	B2	C1	C2	Test Set
ME6 Baseline	en	0.38	0.05	0.29	0.29	0.31	0.47	10% of EVP
ME6 Contextual	en	0.63	0.49	0.45	0.50	0.42	0.60	10% of EVP
ME4 Contextual	en	0.69	0.42	0.53	0.71			10% of EVP
MME4	en, fr	0.66	0.39	0.51	0.70			10% of EVP
Contextual	en, fr	0.66	0.13	0.17	0.05			FEC1
	en, fr	0.60	0.15	0.06	0.06			FEC2
	en, fr	0.47	0.18	0.13	0.12			FEC3
MMEFR4	en, fr	0.65	0.40	0.35	0.71			1% of (EVP + FEC1)
Contextual	en, fr	0.68	0.31	0.51	0.00			FEC2
	en, fr	0.62	0.27	0.30	0.00			FEC3

Table 7: Language support, F1 scores, and test set of trained models

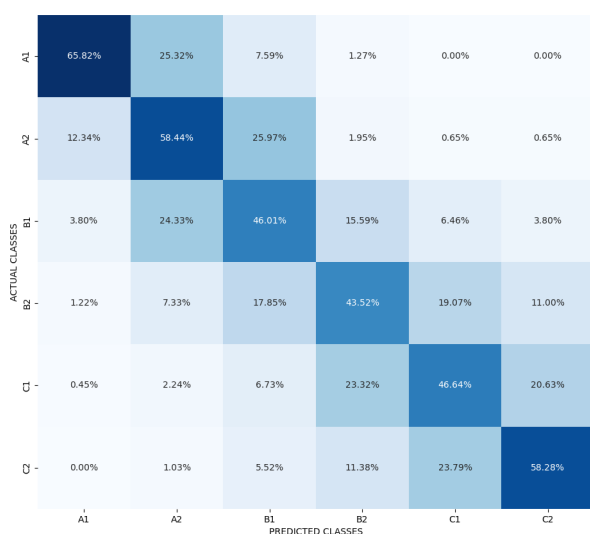


Figure 3: Confusion matrix with normalized scores per class of model ME6 Contextual

complex morphology of the French language. Still, the rich contextual representations encode enough information to allow the model to correctly distinguish and classify, for example, instances of the verb *faire* used as a main verb vs. as an auxiliary.

The early adoption and tests of the vocabulary processing pipeline by Mauril’s team of foreign language teachers highlighted some of the models’ limitations. For example, none of the models (not even the multilingual one) give a special treatment to cognates (sets of words in one of the two languages that have been inherited in direct descent from the other one) which are normally considered easier to acquire. Another concern has been the lack of transparency in the classifier’s predictions, a direct consequence of the dense representations we favoured over the more interpretable linguistic

features. Finally, a contextual classifier may predict different levels for occurrences of the same lexeme in different contexts. Those limitations underline the need for human validation of the output of such systems.

## 7 Conclusion

In this article, we detailed the creation and evaluation of a lexical complexity classifier in French and English, predicting contextually-aware CEFR-based labels for words and multi-word expressions alike. We established a baseline for the six-way lexical classification on the EVP corpus and showed that replacing the representation by statistical features such as frequency for a dense contextual embedding from a masked language model such as BERT achieves a significantly improved accuracy in English and a moderate one in French. The most significant obstacle laying before the creation of an equally performant model in French is the lack of appropriate training data. The ideal corpus would not only contain contextually grounded lexemes, but would reflect productive rather than receptive knowledge of vocabulary.

The utility of a graded lexical classifier goes beyond Mauril’s use case of vocabulary analysis. Such a model may be used in modular text simplification systems to help adjust the level of simplification and adapt it to the user’s competence level. In pipelines for lexical simplification, a CEFR-based classifier might help with the ranking of substitution candidates by providing an estimation of their complexity (in context) (Gooding and Kochmar, 2019b; Aleksandrova and Dufour, 2022). It is also a fine-grained tool for complex word identification and readability analysis.

## References

- Rodrigo Alarcon, Lourdes Moreno, Isabel Segura-Bedmar, and Paloma Martínez. 2019. Lexical simplification approach using easy-to-read resources. *Procesamiento del Lenguaje Natural*, 63(0):95–102.
- Desislava Aleksandrova and Olivier Brochu Dufour. 2022. RCML at TSAR-2022 Shared Task: Lexical simplification with modular substitution candidate ranking. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 259–263.
- David Alfter. 2021. *Exploring natural language processing for single-word and multi-word lexical complexity from a second language learner perspective*. Ph.D. thesis.
- David Alfter, Yuri Bizzoni, Anders Agebjörn, and others. 2016. From distributions to labels: A lexical proficiency analysis using learner corpora. *Proceedings of the joint workshop on NLP for Computer Assisted Language Learning and NLP for Language Acquisition*.
- David Alfter and Elena Volodina. 2018. Towards single word lexical complexity prediction. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 79–88, New Orleans, Louisiana. Association for Computational Linguistics.
- Annette Capel. 2010. A1–B2 vocabulary: insights and issues arising from the English profile wordlists project. *English Profile Journal*, 1.
- Annette Capel. 2012. Completing the English vocabulary profile: C1 and C2 vocabulary. *English Profile Journal*, 3.
- Tom Cobb. 2007. Why & how to use frequency lists to learn words.
- Council of Europe. 2001. *Common European framework of reference for languages: Learning, teaching, assessment*. Cambridge University Press.
- Scott Crossley, Tom Salsbury, and Danielle McNamara. 2010. The development of polysemy and frequency use in English second language speakers. *Lang. Learn.*, 60(3):573–605.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. [BERT: pre-training of deep bidirectional transformers for language understanding](#). *CoRR*, abs/1810.04805.
- Didier FLE. 2004. [Les référentiels](#).
- Thomas François, Mokhtar Boumediene Billami, Núria Gala, and Dephine Bernhard. 2016. Bleu, contusion, ecchymose: tri automatique de synonymes en fonction de leur difficulté de lecture et compréhension. *JEP-TALN-RECITAL*.
- Thomas François, Núria Gala, Patrick Watrin, and Cédric Fairon. 2014. FLELex: a graded lexical resource for French foreign learners. *International conference*.
- Thomas Francois and Patrick Watrin. 2011. On the contribution of MWE-based features to a readability formula for French as a foreign language.
- Núria Gala, Thomas François, Delphine Bernhard, and Cédric Fairon. 2014. Un modèle pour prédire la complexité lexicale et graduer les mots. In *TALN'2014*, pages 91–102.
- Núria Gala, Thomas François, and Cédric Fairon. 2013. Towards a French lexicon with difficulty measures: NLP helping to bridge the gap between traditional dictionaries and specialized lexicons. *eLex-Electronic Lexicography*.
- Garí Soler and Marianna Apidianaki. 2021. Let's play mono-poly: BERT can reveal words' polysemy level and partitionability into senses. *Trans. Assoc. Comput. Linguist.*
- Sian Gooding and Ekaterina Kochmar. 2019a. Complex word identification as a sequence labelling task. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1148–1153, Florence, Italy. Association for Computational Linguistics.
- Sian Gooding and Ekaterina Kochmar. 2019b. Recursive Context-Aware lexical simplification. 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 4853–4863, Hong Kong, China. Association for Computational Linguistics.
- Gouvernement du Québec. 2014. *Agir pour interagir*. Ministère de l'Immigration, de la Diversité et de l'Inclusion.
- Ray Jackendoff. 1995. *The boundaries of the lexicon. Idioms, structural and psychological perspectives*. Hillsdale, NJ: Lawrence Erlbaum.
- Ekaterina Kochmar, Sian Gooding, and Matthew Shardlow. 2020. Detecting multiword expression type helps lexical complexity assessment. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4426–4435, Marseille, France. European Language Resources Association.
- Lété. 2004. MANULEX: une base de données du lexique écrit adressé aux élèves. *Didactique du lexique*.
- Deryle Lonsdale and Yvon Le Bras. 2009. *A Frequency Dictionary of French: Core Vocabulary for Learners*. Routledge.
- William E Nagy. 1995. On the role of context in first- and second-language vocabulary learning. *Center for the Study of Reading Technical Report ; no. 627*.

- Kai North, Marcos Zampieri, and Matthew Shardlow. 2023. Lexical complexity prediction: An overview. *ACM Comput. Surv.*, 55(9):1–42.
- Alice Pintard and Thomas François. 2020. Combining expert knowledge with frequency information to infer CEFR levels for words. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with READING Difficulties (READI)*, pages 85–92, Marseille, France. European Language Resources Association.
- John Platt et al. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.
- Norbert Schmitt, Karen Dunn, Barry O’Sullivan, Laurence Anthony, and Benjamin Kremmel. 2021. Introducing knowledge-based vocabulary lists (KVL). *TESOL j.*, 12(4).
- Burr Settles, Geoffrey T. LaFlair, and Masato Hagiwara. 2020. Machine Learning–Driven language assessment. *Transactions of the Association for Computational Linguistics*, 8:247–263.
- George Kingsley Zipf. 1945. The meaning-frequency relationship of words. *The Journal of general psychology*, 33(2):251–256.

# The NCTE Transcripts: A Dataset of Elementary Math Classroom Transcripts

**Dorottya Demszky**  
Stanford University  
ddemszky@stanford.edu

**Heather Hill**  
Harvard University  
heather\_hill@gse.harvard.edu

## Abstract

Classroom discourse is a core medium of instruction – analyzing it can provide a window into teaching and learning as well as driving the development of new tools for improving instruction. We introduce the largest dataset of mathematics classroom transcripts available to researchers, and demonstrate how this data can help improve instruction. The dataset consists of 1,660 45-60 minute long 4th and 5th grade elementary mathematics observations collected by the National Center for Teacher Effectiveness (NCTE) between 2010-2013. The anonymized transcripts represent data from 317 teachers across 4 school districts that serve largely historically marginalized students. The transcripts come with rich metadata, including turn-level annotations for dialogic discourse moves, classroom observation scores, demographic information, survey responses and student test scores. We demonstrate that our natural language processing model, trained on our turn-level annotations, can learn to identify dialogic discourse moves and these moves are correlated with better classroom observation scores and learning outcomes. This dataset opens up several possibilities for researchers, educators and policymakers to learn about and improve K-12 instruction. The dataset can be found at <https://github.com/ddemszky/classroom-transcript-analysis>.

## 1 Introduction

Improving K-12 mathematics instruction in the aftermath of the Covid-12 pandemic is a major national priority, drawing support from both the U.S. government (U.S. Department of Education)<sup>1</sup> and major foundations (e.g., Gates, Spencer).<sup>2</sup> A key

<sup>1</sup><https://www.ed.gov/news/press-releases/us-department-education-announces-over-220-million-dollars-investments-government-private-and-public-sectors-support-student-recovery>

<sup>2</sup><https://www.spencer.org/news/announcing-covid-19-related-special-grant-cycle>

step in this direction is to measure and facilitate the use of effective mathematics teaching practices, an effort that draws on a long history of research (e.g., Brophy, 1984; Sedova et al., 2019). Instructional measurement has traditionally relied on resource-intensive classroom observation. Recent natural language processing models, trained on manually scored classroom transcripts, enable measuring effective instructional practices in scalable and adaptable ways (Kelly et al., 2018; Suresh et al., 2019; Demszky et al., 2021b; Alic et al., 2022; Hunkins et al., 2022). However, a common barrier to evaluating such measures is the lack of comprehensive data sources that link classroom transcripts to external variables, such as student and teacher demographics and learning outcomes.

To address this, we introduce a dataset of 1,660 U.S. 4th and 5th grade elementary math classroom transcripts collected by the National Center for Teacher Effectiveness (NCTE) between 2010-2013 (Kane et al., 2015). The anonymized transcripts represent data from 317 teachers across 4 school districts serving largely historically marginalized student populations. The transcripts are associated with a wide range of metadata: (i) turn-level annotations for discourse moves, (ii) classroom observation scores, (iii) questionnaires that capture teacher background, beliefs and classroom practices, (iv) student administrative data, (v) questionnaires describing student background and classroom experiences, (vi) value added scores, which estimate teachers' contribution to students' academic performance. To our knowledge, this is the largest dataset of math classroom transcripts with linked outcomes available to researchers.

To illustrate how this data can be used to identify effective instructional practices, we build classifiers for discourse moves and validate these measures by correlating them with instructional outcomes. These discourse moves include on vs off task instruction, teachers' uptake of student ideas (Dem-



szky et al., 2021b), teachers’ focusing questions (Alic et al., 2022) and student reasoning — the latter three of which are indicators of dialogic instruction, where students are active participants of the learning process (Bakhtin, 1981; Nystrand et al., 1997; Wells, 1999; Alexander, 2008).

We show that a RoBERTa classifier can learn to predict these discourse moves with moderate to high accuracy, leaving some room for improvement for future work. Importantly, we find that predictions for all of these discourse moves correlate significantly with classroom observation scores that measure instructional quality, teacher sensitivity, and classroom climate, among other items, while controlling for teacher and classroom covariates. Predictions for dialogic moves (teacher uptake, focusing questions and student reasoning) also show a significant positive correlation with teachers’ value added scores. Taken together, these results demonstrate the value of this dataset for developing measures that can help us understand and facilitate effective instruction, for example, by powering automated feedback tools for teachers (Suresh et al., 2021; Demszky et al., 2021a).

## 2 Related Work

We provide an overview of related corpora and methods pertaining to the computational analysis of classroom discourse.

### 2.1 Related Corpora

There is a wide range conversational datasets available to researchers, capturing phone conversations (Godfrey and Holliman, 1997), task oriented dialogue (Budzianowski et al., 2018), meeting transcripts (Bralely and Murray, 2018), among many others. These datasets can provide valuable insights about social dynamics in conversations, but to understand teaching and learning, we need to capitalize on datasets from the educational domain.

Conversational datasets in the education domain are scarce, due to resource intensiveness of data collection and privacy protections. The Measures of Effective Teaching (MET) dataset (Kane et al., 2013) is the most similar to the NCTE data in terms of availability of outcomes. The MET data contains 2,500 4-9th grade classroom recordings collected between 2009-2011 in six U.S. school districts. The recordings cover a variety of subjects, including English Language Arts and mathematics, and the recordings come with classroom ob-

servations scores, teacher and student demographic data and teacher value added scores. Although this dataset is a rich resource for studying instruction, it is challenging to work with, as it requires a paid subscription and interfacing with the data via a remote server that does not currently support many types of machine learning analyses. Furthermore, the MET data does not include transcripts; although subsets of the data have been transcribed as part of various research projects (Liu and Cohen, 2020; Hunkins et al., 2022), these transcripts are only accessible to these respective research teams.

The TalkMoves dataset (Suresh et al., 2022) is currently the largest publicly available collection of transcripts of U.S. math instruction. The TalkMoves data contains 567 K-12 math classroom transcripts, annotated for talk moves based on accountable talk theory (Michaels et al., 2010) and dialog acts from Switchboard (Jurafsky et al., 1997). The NCTE dataset complements the TalkMoves data in terms of availability of observation scores and outcomes. Other public datasets of educational interactions outside of the K-12 domain include the CIMA tutoring dataset (Stasaski et al., 2020), STEM lectures captured in the DRYAD dataset (Reimer et al., 2016) and the Coursera forum discussion dataset (Rossi and Gnawali, 2014), among others.

### 2.2 Computational Analysis of Educational Interactions

Our detection of discourse moves relates to a long-standing line of work on dialog act classification. The dialog act classification literature focuses on domain-agnostic dialog acts, e.g. acknowledgment, repetition, questioning, etc. In this work, we focus on detecting discourse moves that are indicators of better math instruction, similarly to the TalkMoves project (Suresh et al., 2019). Suresh et al. (2019) train transformer-based classifiers, including BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), on annotations for six accountable talk moves, such as keeping everyone together, getting students to relate, revoicing, pressing for reasoning. The NCTE data comes with annotations for related discourse moves (e.g. revoicing in TalkMoves and uptake in NCTE are similar constructs); predicting accountable talk moves on the NCTE data and studying correlations with outcomes is a promising direction for future work.

To better understand linguistic indicators of teacher effectiveness, Liu and Cohen (2020) ana-

lyze English Language Arts classes in the MET dataset (Kane et al., 2013) using topic models, LIWC (Pennebaker et al., 2001) and an open-ended question classifier. The authors conduct factor analysis on several linguistic features and find that a factor indicating interactive, student centered instruction correlates positively with teachers’ value added scores. Hunkins et al. (2022) annotate transcripts of 156 video clips from 6-8th grade math classrooms in the MET dataset for teacher talk moves that support belonging and inclusivity, such as praise, admonishment, controlling language and learning mindset supportive language. They build a random forest classifier to predict these talk moves, and find that admonishment, for example, has a negative correlation with students’ perception of the classroom environment. Finally, closely related work on the NCTE data has demonstrated the positive correlation between teacher uptake (Demszky et al., 2021b) and focusing questions (Alic et al., 2022) and instructional outcomes. In this work, we expand our correlational analyses to all discourse moves annotated on the NCTE dataset.

### 3 Dataset Description

The dataset consists of 1,660 anonymized transcripts of whole lessons, collected as part of the National Center for Teacher Effectiveness (NCTE) Main Study (Kane et al., 2015).<sup>3</sup> The observations took place between 2010-2013 in 4th and 5th grade elementary math classrooms across four districts serving largely historically marginalized students. In the first two years, teachers and students were assigned to each classroom according to their school’s usual procedure for forming classes. In the third year, the NCTE project team randomly assigned teachers to rosters of students provided by the school.

Table 1 provides key statistics about the transcripts, as well as teacher and student demographics. The majority of teachers in the data are white (65%) and female (84%). Whereas the student body is equally split in terms of gender, the majority are students of color (43% African American, 23% Hispanic/Latinx, 8% Asian) and receive free or reduced lunch (67%). This disparity between student-teacher demographics is in accordance with

<sup>3</sup>Parents and teachers gave consent for the study (Harvard IRB #17768), and for de-identified data to be publicly shared for research.

Transcripts	
# of Transcripts	1660
Year 1	697
Year 2	616
Year 3	347
Avg # Transcripts Per Teacher	5.24 (±2.46)
Avg # of Turns	350 (±186)
Avg % of Teacher Turns	50.2% (±4.8%)
Avg # of Words	5733 (±1782)
Avg % of Teacher Words*	87.7% (±7%)
Teachers	
# of Teachers	317
% Male	16%
% Black	22%
% Asian	3%
% Hispanic/Latinx	3%
% White	65%
Avg # of Years of Experience	10.23 (7.28)
U.grad or Grad Degree in Math	6%
BA in Education	53%
Masters Degree	76%
Students	
# of Students	10,817
Grade	4th (51%) 5th (47%)
% Male	50%
% African American	43%
% Asian	8%
% Hispanic/Latinx	23%
% White	23%
% Free or Reduced Lunch	67%
% Special Education Status	13%
Limited English Proficiency	21%

Table 1: Statistics on transcripts and on teachers and students who are mappable to these transcripts via administrative data. For each demographic, we use the naming convention from Kane et al. (2015). Percentages for student grade levels do not add up to 100% due to missing values.

national statistics<sup>4</sup> (Schaeffer, 2021).

<sup>4</sup><https://nces.ed.gov/>

Variable	Description
Turn-level annotations	Annotations for on vs off task instruction, uptake of student contributions, focusing questions and student reasoning.
Transcript-level observation scores	Observation scores by expert raters using two instruments: CLASS (Pianta et al., 2008) and MQI (Hill et al., 2008).
Student questionnaires	Student survey responses about the classroom experience and their household.
Value-added scores	Teachers' value added scores – i.e., an estimate of teachers' contribution to students' test performance.
Student administrative data	Administrative data on students, e.g. their test scores and demographic information.
Teacher questionnaires	Teacher's self reported information about their background, beliefs and classroom practices.

Table 2: Variables linked to the NCTE transcript data. For full documentation, please refer to Kane et al. (2015).

### 3.1 Transcription & Anonymization

Lessons were captured by ThereNow using its Iris system,<sup>5</sup> which featured three cameras, a lapel microphone worn by teachers, and a bidirectional microphone for capturing student talk. The recordings were transcribed by professional transcribers working under contract to a commercial transcription company. Transcripts are fully anonymized: student and teacher names are replaced with terms like “Student J”, “Teacher” or “Mrs. H”. Inaudible talk, due to classroom noise and far field audio is transcribed as *[Inaudible]*. If the transcriber was unsure of a particular word, they transcribed it within brackets, e.g. *It is a city surrounded by [water]*. Square brackets are also used for other transcriber comments, such as *[crosstalk]*, and *[laughter]*. Almost all teacher talk and the majority of student talk could be transcribed: only 4% of teacher utterances and 21% of student utterances contain an *[Inaudible]* marker. Transcripts contain 5,733 words on average, 87.7% of which are spoken by the teacher.

### 3.2 Linked Variables

The transcript data comes with a uniquely rich source of linked variables, summarized in Table 2. These variables include turn-level annotations for various discourse features, classroom observation scores, demographic information about students and teachers, survey data and value-added scores. Please refer to Kane et al. (2015) for a full docu-

mentation of these variables, except for the turn-level annotations, which we describe below.

**Turn-level annotations.** Table 3 includes examples for each discourse feature, annotated at the turn level. In prior work, experts annotated a sample of 2,348 utterance pairs — exchanges between students and teachers — for on vs off task instruction, teachers' uptake of student ideas (Demszky et al., 2021b) and focusing questions (Alic et al., 2022). The annotation process is described in its respective papers. We include the coding scheme on our Github along with the dataset.

In a separate annotation process, experts coded 2,000 student utterances for student reasoning, a key Common Core aligned student practice. The coding process was based on the MQI classroom observation item “Student Provide Explanations” (Hill et al., 2008). To create a sample for labeling, we (i) hold out half of the transcripts for testing, (ii) from the remaining half, sample 30% of transcripts from the top quartile in terms of their Student Provide Explanations MQI score, and 70% from the rest, (iii) filter out student utterances shorter than 8 words, since they are unlikely to substantive reasoning, (iv) randomly sample up to 5 student utterances from each transcript, to balance representation across transcripts, (v) randomly sample 2,000 student utterances by assigning sampling weights proportionate to classroom diversity: combined percentage of African American and Hispanic/Latinx students in classroom. Each example was randomly assigned to one of two math coaches, who are also experts in the MQI coding instrument. Inter-rater

<sup>5</sup>ThereNow is no longer in business, but their technology is now used by IrisConnect: <https://www.irisconnect.com/uk/products-and-services/video-technology-for-teachers/>

agreement on calibration set (n=200) was 90%.

## 4 Computational Analysis

We illustrate how the NCTE transcript dataset can serve as a valuable resource for developing computational measures of classroom discourse. First, we train models that automatically identify dialogic discourse moves by leveraging the turn-level annotations. We then study how these discourse moves correlate with observation scores and teachers' value added scores.

### 4.1 Pre-Processing Annotations

We binarize annotations for each discourse feature in order to make the classification task consistent across features for the purposes of this paper. We also found that providing teachers with feedback using only their positive examples can be effective (Demszky et al., 2021a), and thus in a first pilot experiment, one may not need to preserve fine-grained distinctions between negative, mediocre and positive examples.<sup>6</sup>

Labels for on vs off task instruction and student reasoning are binary, so we simply consider the majority rater label for these discourse moves. Labels for uptake and focusing questions are on multi-level scales. We binarize them by assigning 1 as a label to examples where the majority of raters selected the top category (“high uptake”, “focusing question”) and 0 to all others.

### 4.2 Supervised Classification

We finetune RoBERTa (Liu et al., 2019) on turn-level annotations for each discourse feature. We run finetuning for 5 epochs, a batch size of 8 x 2 gradient accumulation steps. The choice of this model and parameters were optimal for efficient iteration on a single TitanX GPU; we leave model exploration for future work.

For Student on Task and Student Reasoning, the input to the model was a single student utterance. For Teacher on Task, the input to the model was a single teacher utterance. For High Uptake and Focusing Question, the model input was a student utterance and a subsequent teacher utterance, to match what annotators saw while labeling. We balance labels during training by oversampling the

<sup>6</sup>Using these distinctions in teacher feedback is a promising direction of future work, given that contrasting examples can be an effective pedagogical tools (Schwartz et al., 2011; Sidney et al., 2015).

minority category to represent 50% of labels, as we found this process yields better results.

### 4.3 Regression Analysis

Since our ultimate goal is to improve instruction via classroom discourse analysis, we need to understand if our NLP measures of discourse features indeed correlate with observation scores and student outcomes. To understand this question, we (i) follow the procedure described above to fine-tune classifiers for each discourse on *all* annotations, (ii) predict discourse features for the entire NCTE transcript dataset, (iii) run regressions using classroom observation scores and value added scores as dependent variables.

**Model.** We run a linear regression, clustering standard errors at the teacher level. The models are captured by this equation:

$$y_d = x_f \beta_1 + T \beta_2 + S \beta_3 + \epsilon \quad (1)$$

, where  $y_d$  is a vector representing a dependent variable  $d$ ,  $x_f$  is a vector representing our predictions for a particular discourse feature  $f$ ,  $T$  is a matrix of teacher covariates,  $S$  is a matrix of student covariates,  $\beta_1, \beta_2, \beta_3$  are vectors of unknown parameters to be estimated and  $\epsilon$  is a vector of residuals.

As for discourse features  $f$ , we use discourse moves in Table 3 and also include baseline measures of student talk ratios (% of student words and % of student turns) for comparison. We estimate the effect of these features on six different dependent variables  $d$ , which include five items from observation scores and value added scores.

**Observation scores as dependent variables.** To measure mathematics instructional quality, we use the main holistic item from the MQI instrument, a 5-level rating for lesson quality. The other four variables come from the CLASS scoring instrument: instructional dialogue, teacher sensitivity, teachers' regard for student perspectives and positive classroom climate. We picked these items *a priori*—before conducting the regressions—based on their relevance to dialogic instruction. One could choose other items from these observation protocols to conduct similar analyses.

Since observation scores are linked to transcripts, we first aggregate discourse move predictions to the lesson level. Specifically, we sum discourse feature predictions in each transcript, and divide

Discourse Feature	Example
Student on Task	S: We both have the same number of blue, and red, and yellow.
Teacher on Task	T: Good, find the range. Find the range. Remember it's the span of the least to the greatest number.
Student Reasoning	S: Because if you add ninety-eight hundredths and five hundredths, I think it's going to add up to, like, it's almost – it's going to almost add up to a hundred.
High Uptake	S: I think these are Y axis and the X axis. T: They do. Sometimes they refer to them as X and Y axis. It depends on the type of graph. Okay. You ready?
Focusing Question	S: Four fifths – no, 80 percent. T: How come you can't put it there?

Table 3: Examples for each discourse feature annotated at the turn-level. See Demszky et al. (2021b) and Alic et al. (2022) for more details.

them by the class duration (number of 7.5 minute segments in the transcript<sup>7</sup>).

#### Value added scores as dependent variables.

Each teacher is linked to one value added score per year given that these are based on end-of-the-year standardized test scores. Therefore, we mean-aggregate lesson-level data obtained above to the teacher-year level when conducting regressions with value-added scores.

**Covariates.** We include several covariates for teacher and classroom demographics. We include binary indicators for teacher gender and race/ethnicity and a numerical variable for years of experience. We include variables related to classroom composition in terms of student gender, race, free or reduced lunch status, special education status and limited English proficiency status – see Table 1 for a list of variables.

## 5 Results

**Supervised classification.** Table 4 shows the performance of our supervised classifiers, averaged across five-fold cross validation. The results show that we can train our model to automatically classify each discourse feature with moderate to high accuracy. The model performs best on classifying on vs off task instruction — F1 score (harmonic mean of precision and recall) is .942 and .923 for student and teacher utterances, respectively. The model performs moderately well on higher-inference discourse moves, including Student Reasoning (F1 = .651), High Uptake (F1 = .688), and Focusing Questions (F1 = .501). We expect that model choice and hyperparameter tuning

<sup>7</sup>We do not have consistent timestamps that are more granular than 7.5 minutes.

can improve the performance by 10-20%. However, even the human raters are only able to reach moderate agreement on these measures (Demszky et al., 2021b; Alic et al., 2022) and other analogous ones in classroom observation instruments (Kelly et al., 2020). The moderate human agreement indicates that these measures are subjective, which may set an upper bound to the models' performance.

**Correlation with outcomes.** Table 5 shows the correlation of each discourse feature with outcomes. We also include two baseline discourse features that measure student talk ratios: the percentage of student turns and the percentage of student words in each transcript. We find that **all discourse features** predicted by our classifiers correlate significantly with each classroom observation item measuring instruction quality and classroom climate.

All dialogic discourse moves — High Uptake, Focusing Questions and Student Reasoning — also show a significant positive correlation with teachers' value added scores. Specifically, each additional High Uptake per 7.5 minute segment increases teachers' value added scores by 12% of a standard deviation. Analogically, each additional instance of a Focusing Question and Student Reasoning per 7.5 minute segment increases teachers' value added scores by 23% and 19% of a standard deviation, respectively. Student on Task and Teacher on Task also correlate with marginal significance with value added scores.

Interestingly, baseline measures of student talk percentages do not in themselves correlate with observation scores or value added scores. One exception is student word percentage, which correlates positively with ratings for instructional dialogue.

Measure	Accuracy	Precision	Recall	F1
Student on Task	0.902	0.952	0.931	0.942
Teacher on Task	0.867	0.932	0.914	0.923
Teacher Uptake	0.768	0.719	0.674	0.688
Focusing Question	0.856	0.474	0.538	0.501
Student Reasoning	0.863	0.644	0.666	0.651

Table 4: Performance of RoBERTa on each discourse feature. Values are averages across a 5-fold cross validation.

	Value Added Scores	Math Instruction Quality	Instructional Dialogue	Teacher Sensitivity	Regard for Student Perspectives	Positive Climate
Student on Task	0.038+ (0.020)	<b>0.022*</b> (0.010)	<b>0.032**</b> (0.011)	<b>0.033**</b> (0.008)	<b>0.024**</b> (0.007)	<b>0.036**</b> (0.007)
Teacher on Task	0.038+ (0.020)	<b>0.021*</b> (0.010)	<b>0.030**</b> (0.010)	<b>0.034**</b> (0.008)	<b>0.024**</b> (0.007)	<b>0.035**</b> (0.007)
Teacher Uptake	<b>0.121*</b> (0.050)	<b>0.117**</b> (0.032)	<b>0.083**</b> (0.026)	<b>0.089**</b> (0.019)	<b>0.058**</b> (0.017)	<b>0.079**</b> (0.017)
Focusing Question	<b>0.234*</b> (0.104)	<b>0.233**</b> (0.086)	<b>0.198**</b> (0.072)	<b>0.132**</b> (0.035)	<b>0.164**</b> (0.044)	<b>0.115**</b> (0.036)
Student Reasoning	<b>0.191*</b> (0.091)	<b>0.313**</b> (0.066)	<b>0.246**</b> (0.050)	<b>0.144**</b> (0.031)	<b>0.173**</b> (0.035)	<b>0.120**</b> (0.035)
Student Turn %	1.044 (1.357)	-0.047 (0.528)	0.718 (0.669)	0.214 (0.574)	0.125 (0.485)	-0.172 (0.560)
Student Word %	0.359 (0.792)	0.721+ (0.413)	<b>1.132*</b> (0.541)	0.001 (0.325)	0.469 (0.395)	0.322 (0.387)
Observations	523	1557	1554	1554	1554	1554

Table 5: The correlation of discourse features with outcomes, estimated at the transcript level. Each cell represents coefficients from a separate regression. Standard errors are enclosed in parentheses. Dependent variables are standardized. The \*\*  $p < 0.01$ , \*  $p < 0.05$ , +  $p < 0.1$

## 6 Discussion

We find that our measures of discourse moves, which we identified by consulting research on mathematics instruction, correlate with human raters’ perceptions of lesson quality, and with students’ learning outcomes. These results are significant in multiple ways.

### 6.1 Implications & Significance

The fact that all of our measures correlate with MQI and CLASS observation scores indicate that the automated measures align with expert evaluations of instruction. This is a key result that provides external validation for these automated measures. The positive correlation of teacher value-added scores with measures for teacher uptake, focusing questions and student reasoning suggests that the use of these dialogic talk moves is associated with student learning. Substantively, this finding contributes evidence that classrooms where students are more deeply engaged with mathematical ideas — and where teachers use their students’ mathematical contributions — are more likely to produce better achievement outcomes (O’Connor and Michaels,

1993; Michaels and O’Connor, 2015).

These findings become even more significant in the context of baseline measures of student engagement — percentage of student turns and percentage of student words — which do not show positive correlations with instruction quality and value added scores. These findings add to a collection of mixed results by related work, some of which show positive, some of which show no relationship between student talk time and learning outcomes (see Sedova et al., 2019, for an overview).

### 6.2 Limitations

**Unmeasured covariates.** A range of factors may affect instructional outcomes, only a subset of which could be measured with this data. Making strong claims about the link between discourse moves and instructional outcomes requires experimental validation. For example, the quality of the math task that the students are working may affect the discourse as well as learning outcomes. We can isolate the effect of discourse moves by randomly assigning teachers to learning opportunities that help them improve their use of these moves,

and examining downstream impacts of these new talk moves on student outcomes. (Demszky et al., 2021a) has taken a similar approach successfully in an informal teaching context, but such a study is yet to be done in a K-12 context.

**Generalizability.** Although the NCTE transcript dataset is the largest available dataset of U.S. classroom transcripts, it only captures a tiny fraction of U.S. classrooms and hence there are limitations to its representativeness. The data represents mostly white female teachers working in mid-size to large districts, so it would be valuable to collect new data from other types of districts and a more diverse teacher population. The fact that the data was collected a decade ago may pose limitations to its ongoing relevance; during the period under study (2010-2013), many schools were transitioning toward Common Core-aligned instruction in mathematics but yet lacked high-quality curriculum materials for doing so. That said, research in education reform has long attested to the fact that teaching practices have remained relatively constant over time (Cuban, 1993; Cohen and Mehta, 2017) and that there are strong socio-cultural pressures that maintain the instructional status quo (Cohen, 1988). In general, it is important to carefully validate measures built on the NCTE data on a new domain to ensure that it is representative of the target population.

**Limitations of linked data.** Education research has attested the limitations of standardized assessments in capturing student learning and reasoning (Sussman and Wilson, 2019). Student questionnaires in the NCTE data can provide an alternative perspective on students' experiences and mathematics outcomes but these responses have a lot of missing values, and hence it may not provide robust estimates. Furthermore, understanding equity in instruction is a high priority for our research team and for the field more generally. However, studying equity within this data is challenging, since student speakers are not linked to administrative files containing student background and achievement variables. That said, such speaker-level demographic data is rarely available in instructional contexts, for important ethical reasons, and thus this limitation may encourage researchers to develop measures of instructional equity that leverage classroom-level, instead of speaker-level demographic information.

### 6.3 Ethical Considerations

We outline measures to safeguard students and teachers in this data and the users of it.

**Consent & privacy.** Both parents and teachers gave consent for the de-identified data to be retained and used in future research. It is our highest priority that the identity of teachers and students in the data are kept private. Given that the none of the district and school names are disclosed, and that transcripts are fully de-identified, it is not possible to recover the identity of teachers and students.

**Representation.** As Madaio et al. (2022) point out, applying AI in the educational domains comes with a risk of propagating and exaggerating existing inequities. As we describe in the paper, the data represents a largely low-income, demographically diverse student population. This means that the data can help with creating measures that are representative of low income students, students of color and students who are English language learners who receive the type of instruction captured in this dataset. As we point out above, the data should not be assumed to represent the diversity of identities and experiences of all students and teachers in U.S. classrooms and different forms of instruction. Furthermore, the data was annotated by raters whose demographics are largely representative of teacher demographics in the US<sup>8</sup> (Demszky et al., 2021b), which, just like in this data, does not unfortunately match U.S. student demographics. Rater bias (Campbell and Ronfeldt, 2018) cannot be outruled especially given the subjectivity of these constructs.

**Downstream application.** Users of this data have to agree to never use the data in a way that may cause harm to students and teachers, such as to build tools that discriminate against different groups of students and teachers or to surveil and punish teachers based on their practice. The sole purpose of this data should be to help us understand and facilitate student-centered and equitable instructional practice, and to empower historically marginalized teacher and student populations.

### 6.4 Directions for Future Work

The NCTE dataset opens up numerous directions for future work, some of which we are currently

<sup>8</sup><https://nces.ed.gov/fastfacts/display.asp?id=28>

pursuing. First, one key direction is **building new NLP measures** of instruction. Observation protocols such as MQI, CLASS and the Culturally Responsive Instruction Observation Protocol (CRIOP) (Powell et al., 2016) and related work by Suresh et al. (2022) and Hunkins et al. (2022) can provide inspiration for various discourse features that can be measured in this data. Given the context-dependence of several instructional moves, new measures can incorporate more context beyond a single utterance or exchange between the student and the teacher. It would also be valuable to incorporate lesson-level metadata in the NLP model, e.g. lesson keywords, date, grade level, to create more context-specific measures.

One can also conduct **bottom-up exploration of linguistic patterns** in the data to inform education research. For example, one could create an equity gap measure, leveraging academic outcomes and classroom demographics, and compare transcripts from classrooms with low equity gap with ones from classrooms with high equity gap. Doing so can help identify instructional correlates of equity gap, and help us understand how we can facilitate equitable instructional practices.

Third, since no two education settings are the same, it would be extremely valuable to **complete the NCTE dataset with other educational datasets** from a diverse range of settings, including various school subjects, informal and formal, on-line and in person, group and one-on-one settings, as well as data from multiple regions and countries and from multiple modalities, including text, audio and video. These datasets together can help us understand how effective instruction looks like across teaching contexts and modalities and build measures sensitive to these contextual differences.

Finally, one can **apply what we learn from this data to improve instruction**. Measures built on this data can power automated feedback tools (Suresh et al., 2021; Demszky et al., 2021a) and enable empirically-driven improvements to widely used professional development frameworks (e.g. Gregory et al., 2017, NCTM guides<sup>9</sup>). We hope that this resource can power efforts to address pressing issues in education, such as pandemic learning loss and equity gaps in mathematics instruction.

<sup>9</sup><https://www.nctm.org/pdguides/>

## 7 Conclusion

We introduce a dataset of elementary math classroom transcripts, associated with a rich source of linked variables. We train classifiers on turn-level annotations to predict discourse moves and show that predictions for these moves correlate significantly with observation scores and value added scores. These results demonstrate how the NCTE dataset can serve as a valuable resource for understanding classroom interactions and for powering tools that seek to facilitate instruction.

## Acknowledgments

We thank Shyamoli Sanghi for her assistance with finalizing the dataset. We also thank Jing Liu, Hannah Kleen, Zach Himmelsbach, Zid Mancenido and Dan Jurafsky for the productive conversations and their collaboration. We are grateful to Dan McGinn for help with data sharing and to Christine Kuzdzal and Carol DeFreese for help with annotating student reasoning.

## References

- Robin John Alexander. 2008. *Towards Dialogic Teaching: rethinking classroom talk (4th Edition)*. Dialogos.
- Sterling Alic, Dorottya Demszky, Zid Mancenido, Jing Liu, Heather Hill, and Dan Jurafsky. 2022. Computationally identifying funneling and focusing questions in classroom discourse. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 224–233.
- M. M. Bakhtin. 1981. *The dialogic imagination: four essays*. University of Texas Press.
- McKenzie Braley and Gabriel Murray. 2018. The group affect and performance (gap) corpus. In *Proceedings of the ICMI 2018 Workshop on Group Interaction Frontiers in Technology (GIFT)*.
- Jere E Brophy. 1984. *Teacher behavior and student achievement*. 73. Institute for Research on Teaching, Michigan State University.
- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 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.
- Shanyce L Campbell and Matthew Ronfeldt. 2018. Observational evaluation of teachers: Measuring more than we bargained for? *American Educational Research Journal*, 55(6):1233–1267.



- David K Cohen. 1988. *Teaching practice: Plus ça change*. National Center for Research on Teacher Education East Lansing, MI.
- David K Cohen and Jal D Mehta. 2017. Why reform sometimes succeeds: Understanding the conditions that produce reforms that last. *American Educational Research Journal*, 54(4):644–690.
- Larry Cuban. 1993. *How teachers taught: Constancy and change in American classrooms, 1890-1990*. Teachers College Press.
- Dorottya Demszky, Jing Liu, Heather C Hill, Dan Jurafsky, and Chris Piech. 2021a. Can automated feedback improve teachers’ uptake of student ideas? evidence from a randomized controlled trial in a large-scale online course. edworkingpaper no. 21-483. *Annenberg Institute for School Reform at Brown University*.
- Dorottya Demszky, Jing Liu, Zid Mancenido, Julie Cohen, Heather Hill, Dan Jurafsky, and Tatsunori B Hashimoto. 2021b. Measuring conversational uptake: A case study on student-teacher interactions. 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 1638–1653.
- 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.
- John J Godfrey and Edward Holliman. 1997. Switchboard-1 release 2. *Linguistic Data Consortium, Philadelphia*, 926:927.
- A Gregory, E Ruzek, CA Hafen, A Yee Mikami, JP Allen, and RC Pianta. 2017. My teaching partner-secondary: A video-based coaching model. *Theory into practice*, 56(1):38–45.
- Heather C Hill, Merrie L Blunk, Charalambos Y Charalambous, Jennifer M Lewis, Geoffrey C Phelps, Laurie Sleep, and Deborah Loewenberg Ball. 2008. Mathematical knowledge for teaching and the mathematical quality of instruction: An exploratory study. *Cognition and instruction*, 26(4):430–511.
- Nicholas Hunkins, Sean Kelly, and Sidney D’Mello. 2022. “beautiful work, you’re rock stars!”: Teacher analytics to uncover discourse that supports or undermines student motivation, identity, and belonging in classrooms. In *LAK22: 12th International Learning Analytics and Knowledge Conference*, pages 230–238.
- Daniel Jurafsky, Elizabeth Shriberg, and Debra Bisasca. 1997. Switchboard SWBD-DAMSL Labeling Project Coder’s Manual, Draft 13. Technical Report 97-02, University of Colorado Institute of Cognitive Science.
- T Kane, H Hill, and D Staiger. 2015. National center for teacher effectiveness main study. icpsr36095-v2.
- Thomas J Kane, Daniel F McCaffrey, Trey Miller, and Douglas O Staiger. 2013. Have we identified effective teachers? validating measures of effective teaching using random assignment. research paper. met project. *Bill & Melinda Gates Foundation*.
- Sean Kelly, Robert Bringe, Esteban Aucejo, and Jane Cooley Fruehwirth. 2020. Using global observation protocols to inform research on teaching effectiveness and school improvement: Strengths and emerging limitations. *Education Policy Analysis Archives*, 28:62.
- Sean Kelly, Andrew M Olney, Patrick Donnelly, Martin Nystrand, and Sidney K D’Mello. 2018. Automatically measuring question authenticity in real-world classrooms. *Educational Researcher*, 47(7):451–464.
- Jing Liu and Julie Cohen. 2020. Measuring teaching practices at scale: A novel application of text-as-data methods. *EdWorking PaperNo*, pages 20–239.
- 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*.
- Michael Madaio, Su Lin Blodgett, Elijah Mayfield, and Ezekiel Dixon-Román. 2022. Beyond “fairness”: Structural (in) justice lenses on ai for education. In *The Ethics of Artificial Intelligence in Education*, pages 203–239. Routledge.
- Sarah Michaels and Catherine O’Connor. 2015. Conceptualizing talk moves as tools: Professional development approaches for academically productive discussion. *Socializing intelligence through talk and dialogue*, 347:362.
- Sarah Michaels, Mary Catherine O’Connor, Megan Williams Hall, and Lauren B Resnick. 2010. Accountable talk sourcebook: For classroom conversation that works. *Pittsburgh, PA: University of Pittsburgh Institute for Learning*.
- Martin Nystrand, Adam Gamoran, Robert Kachur, and Catherine Prendergast. 1997. *Opening dialogue*. New York: Teachers College Press.
- Mary C O’Connor and Sarah Michaels. 1993. Aligning academic task and participation status through revoicing: Analysis of a classroom discourse strategy. *Anthropology & Education Quarterly*, 24(4):318–335.

- James W Pennebaker, Martha E Francis, and Roger J Booth. 2001. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001.
- Robert C Pianta, Karen M La Paro, and Bridget K Hamre. 2008. *Classroom Assessment Scoring System: Manual K-3*. Paul H Brookes Publishing.
- Rebecca Powell, Susan Chambers Cantrell, Victor Malojuvera, and Pamela Correll. 2016. Operationalizing culturally responsive instruction: Preliminary findings of criop research. *Teachers College Record*, 118(1):1–46.
- Lynn C Reimer, Katerina Schenke, Tutrang Nguyen, Diane K O’ Dowd, Thurston Domina, and Mark Warschauer. 2016. Evaluating promising practices in undergraduate stem lecture courses. *RSF: The Russell Sage Foundation Journal of the Social Sciences*, 2(1):212–233.
- Lorenzo A. Rossi and Omprakash Gnawali. 2014. Language Independent Analysis and Classification of Discussion Threads in Coursera MOOC Forums. In *Proceedings of the IEEE International Conference on Information Reuse and Integration (IRI 2014)*.
- Katherine Schaeffer. 2021. [America’s public school teachers are far less racially and ethnically diverse than their students.](#)
- Daniel L Schwartz, Catherine C Chase, Marily A Oppezzo, and Doris B Chin. 2011. Practicing versus inventing with contrasting cases: The effects of telling first on learning and transfer. *Journal of educational psychology*, 103(4):759.
- Klara Sedova, Martin Sedlacek, Roman Svaricek, Martin Majcik, Jana Navratilova, Anna Drexlerova, Jakub Kychler, and Zuzana Salamounova. 2019. Do those who talk more learn more? the relationship between student classroom talk and student achievement. *Learning and instruction*, 63:101217.
- Pooja G Sidney, Shanta Hattikudur, and Martha W Alibali. 2015. How do contrasting cases and self-explanation promote learning? evidence from fraction division. *Learning and Instruction*, 40:29–38.
- Katherine Stasaski, Kimberly Kao, and Marti A. Hearst. 2020. [CIMA: A large open access dialogue dataset for tutoring.](#) In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–64, Seattle, WA, USA → Online. Association for Computational Linguistics.
- Abhijit Suresh, Jennifer Jacobs, Charis Clevenger, Vivian Lai, Chenhao Tan, James H Martin, and Tamara Sumner. 2021. Using ai to promote equitable classroom discussions: The talkmoves application. In *International Conference on Artificial Intelligence in Education*, pages 344–348. Springer.
- Abhijit Suresh, Jennifer Jacobs, Charis Harty, Margaret Perkoff, James H Martin, and Tamara Sumner. 2022. The talkmoves dataset: K-12 mathematics lesson transcripts annotated for teacher and student discursive moves. *arXiv preprint arXiv:2204.09652*.
- Abhijit Suresh, Tamara Sumner, Jennifer Jacobs, Bill Foland, and Wayne Ward. 2019. Automating analysis and feedback to improve mathematics teachers’ classroom discourse. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 9721–9728.
- Joshua Sussman and Mark R Wilson. 2019. The use and validity of standardized achievement tests for evaluating new curricular interventions in mathematics and science. *American Journal of Evaluation*, 40(2):190–213.
- Gordon Wells. 1999. *Dialogic inquiry: Towards a socio-cultural practice and theory of education*. Cambridge University Press.

# Auto-req: Automatic detection of pre-requisite dependencies between academic videos

Rushil Thareja, Ritik Garg, Shiva Baghel,  
Deep Dwivedi, Mukesh Mohania, Ritvik Kulshrestha

Extramarks Education India Pvt. Ltd.

FirstName.LastName@extramarks.com

## Abstract

Online learning platforms offer a wealth of educational material, but as the amount of content on these platforms grows, students may struggle to determine the most efficient order in which to cover the material to achieve a particular learning objective. In this paper, we propose a feature-based method for identifying pre-requisite dependencies between academic videos. Our approach involves using a transcript engine with a language model to transcribe domain-specific terms and then extracting novel similarity-based features to determine pre-requisite dependencies between video transcripts. This approach succeeds due to the development of a novel corpus of K-12 academic text, which was created using a proposed feature-based document parser. We evaluate our method on hand-annotated datasets for transcript extraction, video pre-requisites determination, and textbook parsing, which we have released. Our method for pre-requisite edge determination shows significant improvement (+4.7%-10.24% F1-score) compared to existing methods.

## 1 Introduction

In many online learning platforms, academic videos that cover specific concepts are included in the curriculum. These videos cover certain "academic concepts," which are key ideas that are conveyed in the learning material. These fine-grained concepts aid students in understanding the learning content more effectively and achieving their core learning objectives. The prerequisite dependencies between these concepts, which pertain to the order in which they should be covered, are crucial for both educators and learners. They assist educators in curriculum planning and creating better learning pathways for students. With the increasing reliance on online learning platforms, there is a vast amount of academic content that requires proper organization into dependency graphs to aid in indexing

for smart search capabilities and providing defined learning paths for students. Research has shown that organizing content in this manner has significant benefits for learning, even in offline settings. A meta-analysis of 55 studies involving over 5,000 participants found that students who use concept maps for their daily studies were able to learn more in the same amount of time (Nesbit and Adesope, 2006).

Although learning content is organized in textbooks and MOOCs, the creation of dependency graphs for academic videos serves to extend this organization, enabling us to identify only the relevant and required content for a specific learning objective based on prerequisite relationships. Such a system allows us to recommend personalized learning pathways to users, fostering efficient and effective coverage of specific academic concepts. This tailored approach enhances students' educational experiences and promotes better understanding of the subject matter. Moreover, it saves time for the student by ensuring that all required concepts or skills are covered before viewing content related to the desired academic concept. In this study, we propose a two-stage methodology for identifying prerequisite relationships among academic videos. The process begins with transcribing videos utilizing a speech-to-text model, combined with a language model specifically trained on a K-12 domain corpus. Subsequently, we extract innovative similarity-based features from these transcripts to determine the prerequisite connections.

The features employed in our study have been meticulously designed with the guidance of expert educators in the respective domain. These features utilize several similarity-based factors between two videos to identify pre-requisite dependencies. These factors include similarities between titles, content, and taxonomy. We also use keyphrase extraction algorithms to identify the topics covered in the transcripts and then compare the similarity

between them. Our work introduces the use of extracted keyphrase-based similarity for this task, contributing a novel approach to this research domain. Once the features are extracted we use models such as LGBM (Ke et al., 2017), Random Forest (Breiman, 2001), and ExtraTrees (Geurts et al., 2006) to predict prerequisite dependencies. Our approach for identifying prerequisite relationships among educational videos demonstrates superior performance compared to existing benchmarks.

To evaluate our pipeline, we used a hand-labeled dataset of K-12 academic videos with annotated pre-requisite edges. We introduced a novel feature-based PDF document parser that extracts a K-12 text corpus which ensures correct transcription of domain-specific terminologies and extraction of accurate semantic similarity-based features that take into account the contextual meaning of such terms. This tool extracts a hierarchical and well-organized corpus of K-12 academic text from core curriculum textbooks, strengthening the resilience and effectiveness of both pipeline stages when addressing technical vocabulary.

The primary contributions of our research can be enumerated as follows:

- A method to extract transcripts from academic videos by using a text-to-speech model such as Wav2Vec2 (Baevski et al., 2020) along with a language model built from a corpus of K12 academic content.
- A novel set of similarity-based features that can predict prerequisite edges between academic videos.
- A method to parse academic PDF textbooks using novel layout-based features to extract hierarchical learning taxonomies and content.
- We introduce the following datasets:
  - A hand-labeled dataset of over 2797 prerequisite edges between academic videos annotated by domain expert teachers.
  - Extracted transcripts using various methods and ground truth transcripts for a randomly selected subset of videos available in the public domain.
  - Hand-labelled textbooks parsed with all section headers, text body, and chapter names, as well as an object detection textbook page image dataset, with bounding boxes labeled on all instances of section headers.

The datasets are released at <https://bit.ly/412WkQp> and a demo for the generated prerequisite edges can be found at <https://bit.ly/3VrzMYL>.

## 2 Current work

Our end-to-end pipeline to identify prerequisite dependencies between academic videos is novel. However, the sub-problems, such as transcript extraction, prerequisite edge detection, and parsing textbook PDFs have been well-studied in the literature.

### 2.1 Transcript extraction

Speech-to-Text Recognition (STR) technology is widely used in the online learning domain. Previous studies have shown that students, especially those with learning disabilities, can greatly benefit from transcripts of learning content (Leibold and Buss, 2019). With an increase in the availability of large-scale datasets and newer deep-learning algorithms, many different methods have shown great performance on this task. End-to-end sequence-to-sequence (S2S) modeling using RNN-based, Transformer-based, and Conformer based models are often used for this task (Wang et al., 2020). Newer methods such as Wav2Vec2 (Baevski et al., 2020) have achieved great performance by masking speech input in the latent space and solving a contrastive task defined over a quantization of the latent representations which are jointly learned. This model trained on the librispeech automatic speech recognition (ASR) dataset (Panayotov et al., 2015) has found wide adoption for speech-to-text tasks. We augment the Wav2Vec2 speech model with a 5-gram n-gram language model trained on a corpus of K-12 academic textbooks.

### 2.2 Pre-req edge identification

Identification of prerequisite relations between academic concepts has been a subject of study for decades. Teachers and curriculum planners have extensively utilized this knowledge to determine the order in which chapters are organized in conventional learning textbooks and to guide students in covering the syllabus efficiently (Novak, 1990). However, recent data-driven approaches have facilitated the automated identification of prerequisites, resulting in enhanced performance and the emergence of new research avenues. One example is the information-theoretic approach proposed by

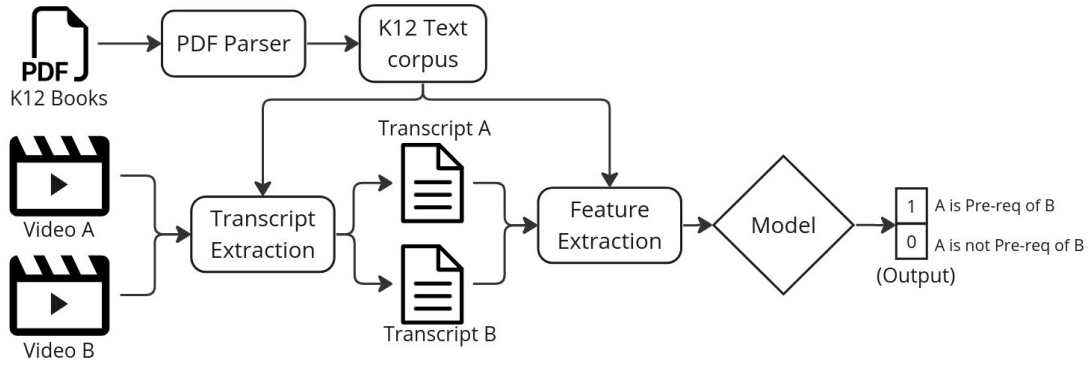


Figure 1: End-to-End system architecture

(Gordon et al., 2016). External knowledge bases, such as Wikipedia, have also been extensively employed. Liang et al. (2019) utilizes active learning on hand-crafted features (Liang et al., 2018b), while Sayyadiharikandeh et al. (2019) leverages Wiki click-stream-based features for prerequisite detection. Additionally, incorporating features similar to those employed in (Liang et al., 2018a), along with Long Short-Term Memory (LSTM) networks, has demonstrated strong performance as reported in (Miaschi et al., 2019). However, finding exact Wikipedia articles for domain-specific academic concepts is an error-prone process with poor results from direct search. Therefore, in our method, we avoid this mapping and find relevant features from the videos themselves. Recently, some methods have been developed to explore the determination of prerequisites between any two textual documents from different domains, including video transcripts, Wikipedia, etc. One such method, leverages aggregated fast-text word embeddings (Bojanowski et al., 2017) for effective prediction of prerequisites (Gaspiretti, 2022). Furthermore, graph-based deep learning methods have also been explored (Li et al., 2019), but these methods tend to require a large amount of training data and may have limited real-world performance.

### 2.3 Parsing Academic Textbook PDFs

PDF parsing is a well-researched issue, historically addressed using rule-based techniques to extract data from documents’ layouts (Mao et al., 2003). Many recent tools use Conditional Random Fields (CRFs) which are undirected graphical models trained to maximize a conditional probability that can be used to segment and label sequence data (Singh et al., 2016).

Additionally, it is possible to treat PDFs as im-

ages and perform text detection and extraction to extract the content. Deep learning computer vision methods have been found to be useful in this regard. For example, Siegel et al. (2018) utilized a modified version of the *ResNet101* network to extract figures and captions from scientific documents. Architectures such as *U-net* (Ronneberger et al., 2015) has also been utilized for performing text body identification (Stahl et al., 2018). Deep learning methods are also effective for finding tables, headers, or citations in PDF files, treating it as an object detection problem. Huang et al. (2019) uses *Yolo* (Redmon et al., 2016) architecture to find tables in PDF files. However, it is important to note that most current work focuses on parsing research papers, and work on academic textbooks is limited.

## 3 Methodology

In this section, we present a comprehensive explanation of the two-stage pipeline used for identifying prerequisite edges between academic videos as shown in Figure 1. The pipeline comprises a transcript extraction stage, followed by a feature extraction and classification stage for prerequisite edge detection. Additionally, the pipeline requires a corpus of academic text obtained from academic textbooks. To fulfill this requirement, we have developed our own academic textbook parser.

### 3.1 Transcript Extraction

The first step in this process is to create a language model that can be used alongside the Wav2Vec2 speech model to improve the transcription of domain-specific terminologies. In order to create this language model, we use our corpus of academic K-12 text. This corpus contains parsed data from classes 9th to 12th for science and math subjects. To create a generic academic video tran-

scriber, we use all textual data from this corpus. However, for a specific class and subject video transcription, it is possible to query data for only that use case and train the language model accordingly. We create a 5-gram n-gram language model using the KenLM method (Heafield, 2011). KenLM performs interpolated modified Kneser Ney Smoothing for estimating the n-gram probabilities (Kneser and Ney, 1995). This model is used to form the decoder, which is combined with the processor’s tokenizer and feature extractor to form the *Wav2Vec2 processor with language model*. We use this *processor* on the output of the Wav2Vec2 Large 960h model trained on the librispeech ASR dataset (Panayotov et al., 2015) to extract transcripts. The fine tuned language model aids the decoding process in Wav2Vec2 by providing context, which adjusts the prediction of the next token in the sequence based on the sequence of previously predicted tokens, thereby enhancing the linguistic coherence of the transcriptions.

However, in order to process MP4 videos through this pipeline, we must first extract audio in the required format. Audio is extracted and saved as an MP3 file. Then, this MP3 file is re-sampled at 16 kHz (the frequency used by the Wav2Vec2 model). Also, as the model only works well with mono-audio, we check if the audio is in stereo format and convert it into mono-audio if required. We use FFmpeg tool (Tomar, 2006) to perform this processing. Finally, the processed audio is saved as WAV files that can be passed into the model to extract transcripts.

### 3.2 Pre-requisite Edge Detection

The problem of finding prerequisites between academic videos is formulated as follows. An academic video corpus of an online learning platform can be represented by  $n$  videos, denoted as  $C = \{V_1, \dots, V_i, \dots, V_n\}$  (1), where each  $V_i$  is one academic learning video. Each video  $V_i$  can be further represented as  $V_i = \{Transcript, Title, Taxonomy, Extracted Phrases\}$  (2).

**Transcript** is the document of video text of the form  $Transcript = (s_1 \dots s_i \dots s_{|V|})$  (3), where  $s_i$  is the  $i^{th}$  sentence of the video text.

**Title** is the heading of the video, which is typically the academic concept that the video covers.

**Taxonomy** is a tuple associated with each video of the form:  $(su, cl, ch, to, st)$  (4) where  $su \in Su, cl \in Cl, ch \in Ch, to \in To, st \in St$  where

the set of all subjects is represented as  $Su$ , the set of all classes as  $Cl$ , the set of all chapters as  $Ch$ , the set of all topics as  $To$ , and the set of all subtopics as  $St$ . All sets,  $Su, Cl, Ch, To$ , and  $St$  pertain to the K12 curriculum. Furthermore, in this paper, we use a subset of  $Su$  and  $Cl$  as follows:  $Su = \{Science, Mathematics, Physics, Biology, Chemistry\}$  and  $Cl = \{x \mid x \in \mathbb{Z}, 6 \leq x \leq 12\}$ .

**Extracted Phrases** is an ordered set, denoted as  $\{p_i \mid i \in \mathbb{N}, 1 \leq i \leq m\}$  (5), comprising of phrases extracted from the Transcript of  $V_i$  (3) using TextRank (Mihalcea and Tarau, 2004). Here,  $m$  represents the total number of extracted phrases, and  $p_i$  denotes the  $i^{th}$  phrase.  $p_i$  is ranked higher than  $p_j$  if  $i < j$ . We opted for TextRank for keyword extraction due to its unsupervised, graph-based nature, which enables it to effectively capture contextual and semantic relationships within the diverse and complex language used in academic video transcripts. Its simplicity and versatility across domains also ensured it could efficiently handle our broad range of data.

Based on these definitions, the problem of finding prerequisites between academic videos in corpus  $C$  (1) can be represented by a function  $F : C^2 \rightarrow \{0, 1\}$ , where :

$$F(\langle a, b \rangle) = \begin{cases} 1 & \text{if } a \text{ is prerequisite of } b \\ 0 & \text{if } a \text{ is not prerequisite of } b \end{cases} \quad (6)$$

and where  $\langle a, b \rangle$  is a video pair (7),  $a, b \in C$  (1). Given this video pair  $\langle a, b \rangle$ , we can extract a set of similarity-based features from their content (2). Let  $(Tr_a, Ti_a, Ta_a, E_a), (Tr_b, Ti_b, Ta_b, E_b)$  (8) be the *transcripts, titles, taxonomies* and *extracted phrases* of videos  $a$  and  $b$ , respectively. In order to find similarity-based features between these, we define a set:

$$content\ pair = \left\{ (x, y) \mid \begin{array}{l} x \in Tr_a, Ti_a, Ta_a, E_a \\ y \in Tr_b, Ti_b, Ta_b, E_b \end{array} \right\} \quad (9)$$

We prune the set *content pair* manually to remove repeated and unnecessary pairs, and then define a function  $S : content\ pair \rightarrow \mathbb{R}$  (10) that computes the similarity between each pair of corresponding elements of the two videos.

Let  $f_i$  be one possible value generated by  $S$ , we take all these possible values together to form the final feature vector  $k = (f_1, f_2, \dots, f_n)$ . These features can then be used to learn the function  $F : C^2 \rightarrow \{0, 1\}$  (6) using a supervised learning algorithm.

### 3.2.1 Calculating Similarity

For calculating the similarity as part of the function  $S$  (10) described above, we use the following approach: We employ two fine-tuned models, *Word2Vec Skip-Gram* (Mikolov et al., 2013), pre-trained on 100B Google News words and fine-tuned with a lock-factor of 0.2 for 5 epochs on our K-12 corpus, and *FastText (FT)* (Bojanowski et al., 2017), also fine-tuned on the same corpus. *Word2Vec* is utilized for phrases with less than 5 words; *FT* for longer phrases. For *Word2Vec*, embeddings are averaged to obtain a 300-dimensional vector, while *FT* directly generates sentence-level embeddings. Cosine similarity is computed between the 300-dimensional vectors to determine similarity scores, with -1 indicating complete dissimilarity and 1 representing identical inputs.

We opted for *Word2Vec* and *FT*, over transformer models, for their computational efficiency and simplicity, given our large transcript dataset. *Word2Vec* was chosen due to its strength in handling common words, while *FT* was selected for its speed and reduced out-of-vocabulary issue, which is particularly useful for longer phrases. Despite the embeddings being in different spaces, the similarity computation remains consistent as we use *Word2Vec* for shorter phrases and *FT* for longer ones, ensuring comparable similarity scores across phrase lengths.

### 3.2.2 Features Extracted

The following features are extracted for each video pair  $\langle a, b \rangle$  (7):

- **Title similarity:** the similarity between the titles of the two videos  $Ti_a, Ti_b$  (8), is expected to be higher if the videos occur in a linked context in the K-12 corpus, suggesting that they have pre-requisite dependencies.
- **Taxonomy Similarity:** Chapter- and subject-based information is vital for determining the prerequisite order of videos. Hence, we calculate the similarity as described above between the taxonomies of two videos  $Ta_a, Ta_b$  (8).
- **Title and Transcript similarity:** The title of a video appearing in the transcript of another video can be utilized to find dependencies. Therefore, we find similarity between the Title and Transcript  $Ti_a, Tr_b$  and  $Ti_b, Tr_a$  (8):
  - Simple count of Title and its subsentences in the Transcript.
  - Sum of similarities between Title and all phrases in the Transcript i.e for  $Ti_a, Tr_b$  we compute

$$\sum_{i=1}^{|V_b|} \sum_j^{phrases(s_i)} S(Ti_a, j) \quad (10)$$

where,  $phrases(s_i)$  represents the word phrases in the sentence  $s_i$  and not the extracted phrases using textrank.

- Cosine similarity between the TF-IDF vectors of Title and Transcript.

Additionally, we apply this process to the first 500 characters of the Transcript, as these initial sentences often contain crucial information that indicates prerequisite relationships (Liang et al., 2018a).

- **Title and extracted phrases similarity:** The title of one video occurring as an important topic in another video can indicate that it is a prerequisite. Thus, we calculate the similarity between  $Ti_a, E_b$  and  $Ti_b, E_a$  (8):

$$- \sum_{i=1}^{|E_b|} S(Ti_a, p_i) \text{ where } p_i \in E_b \text{ and } \sum_{i=1}^{|E_a|} S(Ti_b, q_i) \text{ where } q_i \in E_a.$$

- List of instances where the similarity exceeds specific thresholds:

$$\{p_i \in E_b | S(Ti_a, p_i) > t\} \text{ and } \{q_i \in E_a | S(Ti_b, q_i) > t\},$$

where  $t \in \{0.1, 0.2, \dots, 0.9\}$  (11)

- **Title and taxonomy similarity:** We compute  $S(Ti_a, j)$  where  $j \in Ta_b$  and  $S(Ti_b, l)$  where  $l \in Ta_a$  (4) to take into account the relatedness of the video title  $Ti$  with the *subject*, *chapter*, *topic* or *sub-topics* in the taxonomy  $Ta$  of the other video.
- **Similarity between extracted phrases:** For each phrase  $p_i \in E'_a$ , where  $E'_a$  denotes the top 10 extracted phrases in  $E_a$ , we find the similarity with the extracted phrases in  $E_b$  (5), and then sum these similarities while multiplying with the weight  $w_i$ :

$$w_i \sum_{p_j \in E_b} S(p_i, p_j) \text{ where } w_i = \frac{1}{\lambda^i}$$

and  $i \in \mathbb{N} : 1 \leq i \leq 10$ . We obtained the best results when  $\lambda = 1.1$ . The motivation behind the weighting parameter arises from the notion that higher-ranked phrases tend to be of greater importance or relevance for prerequisite determination. By incorporating this weighting scheme, we assign more weight to the phrases that are ranked higher, hence magnifying their influence on the similarity score.

- **Similarity between video content:** To calculate the overall similarity between the two transcripts, we utilize cosine similarity between their TF-IDF vectors, treating them as two independent textual documents.

For calculating similarity between two large video transcripts, we use TF-IDF due to its computational efficiency and its capacity to detect recurring themes. TF-IDF, when combined with cosine similarity, enables us to compute the overall resemblance between transcripts, irrespective of their length. This makes it a practical solution for identifying textual similarities in extensive video transcripts.

The aforementioned features result in a feature vector of size 316. Additionally, we append a 665-length Bag of Words (BOW) vector, representing the combined titles of the two videos in the format "<Title of Video A> <Space> <Title of Video B>". This yields a combined feature vector of size 981, which is used to train our models in a supervised setting. We evaluated the performance of 36 widely-used machine learning models for all supervised tasks in this study, and present the results of the models that demonstrated superior performance.

### 3.3 Parsing Academic Textbook PDFs

Previously, it was demonstrated that a hierarchically organized and clean K-12 academic corpus is essential for both transcript extraction and prerequisite edge determination. To accomplish this, we have created a collection of academic textbook PDFs that are publicly available<sup>1</sup>. We have selected PDF textbooks in the science, physics, chemistry, biology, and mathematics domains for classes 9th through 12th. Initially, these PDFs are converted to XML using PDF2XML (Peng and Zhang, 2004). Following this, we classify each font into one of three text classes: *chapter names*, *section* or *subsection headers*, and *text body*, based on the following features:

- **Font frequency and size:** *Chapter names* and *section headers* use fonts that are larger and occur less frequently than the general text, making their font occurrence frequency and size distinct from the general text.
- **Font location and page occurrence:** *Chapter names* and *section headers* are positioned at

the top of the page, and *chapter names* occur earlier in the overall text. This allows the use of statistical measures of font average location and page number, to distinguish between different *text classes*.

- **Color:** *Section headers* and *chapter names* frequently use distinct colors. We calculate Euclidean color distance (12) between font color and black and white colors to quantify the font color’s uniqueness compared to the page’s most common colors.

$$dist(C_1, C_2) = \sqrt{(r_1 - r_2)^2 + (g_1 - g_2)^2 + (b_1 - b_2)^2} \quad (12)$$

where  $C_1$  and  $C_2$  represent RGB color values  $[r_1, g_1, b_1]$  and  $[r_2, g_2, b_2]$  respectively.

- **Line width and section numbers:** Section numbers (13) can distinguish *section headers* from other *text classes*. Additionally, *chapter names* tend to have a narrower average line width.

$$Sectionno. = x.y.z \text{ or } x.y, \text{ where } x, y, z \in N \quad (13)$$

Upon extracting the features, a machine learning model classifies each font into three text classes, assigning a class to each text line based on its font. Following the extraction of academic content, section and chapter names, section numbers in headers are utilized to derive the taxonomy. The extracted textual data and its hierarchical structure are included in the released datasets.

## 4 Dataset

### 4.1 Transcript dataset

To showcase the efficacy of our proposed *Wav2Vec2* speech model combined with the language model trained on our K-12 corpus, we assembled a dataset comprising five random academic videos in the science and math domains from YouTube. We provide ground truth subtitles for these videos, alongside subtitles extracted by our algorithm and other benchmarks for comparison.

### 4.2 VID-REQ pre-requisite dataset

To assess our approach, we introduce *Vid-Req*, a large-scale video prerequisite edge dataset. We initially gathered over 1,500 animated academic videos covering science, mathematics, chemistry, physics, and biology for grades 6 through 12 from *Extramarks* a leading *EdTech* company. On average, each video encompasses 418 words. However, these videos resulted in 1,124,250 distinct

<sup>1</sup>NCERT website



video pairs ( ${}^{1500}C_2$ ), which was an overwhelming amount for labeling. Consequently, we selectively choose videos based on a specific criterion to reduce the dataset to a more manageable size. For this purpose, we firstly find chapter-level prerequisites and formulate the set  $CP = \{(ch_1, ch_2) | ch_1 \text{ is a prerequisite of } ch_2\}$  where  $ch_1, ch_2$  are chapters. Using  $CP$ , we form the potential video prerequisites set  $PVP = \{(a, b) | a, b \in C, (ch_a, ch_b) \in CP, ch_a \in Ta_a, ch_b \in Ta_b\}$  (1,4,9). Then, we prune the set  $PVP$  to form  $PVP' = \{(a, b) | S(Ti_a, Ti_b) > 0.7, (a, b) \in PVP\}$ . This set comprises 2,797 edges that we have hand-labeled, of which 1,684 are labeled as 0 (non-prerequisite edges) and 1,113 as 1 (prerequisite edges).

Figure 2 displays the pre-requisite edge statistics for the entire dataset, including label 0 (not pre-requisites) and label 1 (pre-requisite edges) on the left, and only label 1 on the right. The figure shows that science-to-science edges are most frequent in the total dataset (n=1167), but in the label=1 set (n=455), mathematics-to-mathematics edges prevail (n=470). While mathematics appears as a pre-requisite for all subjects in the full edge set, it only acts as an actual pre-requisite for itself and science. Science remains a pre-requisite for other subjects, with most pre-requisite edges leading to physics, biology and chemistry (n=61,23,20).

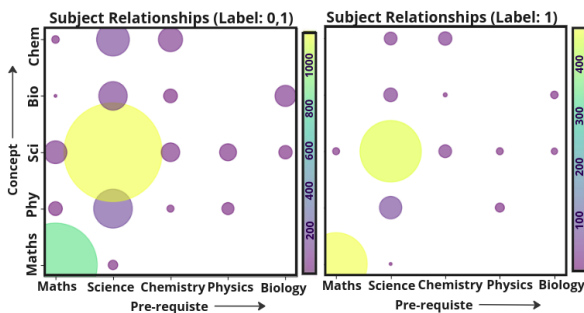


Figure 2: The subject relationship in VID-REQ with all edges on the left and only those labeled 1 on the right

#### 4.2.1 Annotation Process

Multiple experienced teachers were invited and assigned to their preferred subjects, with at least three teachers per subject. These domain experts annotated video pairs, determining if video "B" had a prerequisite video "A" by assigning binary labels (1: A is a prerequisite of B, 0: A is not a prerequisite of B) and also assigned a unique taxonomy from the set of taxonomies extracted from K12 text-

books parsed using our PDF parser to each video. Teachers viewed the videos thoroughly before annotating and provided well-informed judgments and reasons. The relationship is non-symmetric. After annotating 2797 video edges, Cohen's Kappa coefficient (0.64) confirmed substantial agreement among annotators. These final annotations served as ground truth labels for model training.

#### 4.3 Academic textbooks dataset

We generated a training dataset for PDF parsing by downloading 26 textbooks from<sup>2</sup> and converting them to XML using PDF2XML. These textbooks span various subjects and classes, covering 662 unique fonts for *chapter names* (n=53), *text body* (n=563), and *section name* (n=46) text classes, hand-labeled by expert academicians. The model trained on this dataset was used to parse 189 PDFs for subjects like science, math, chemistry, biology, and physics for classes 9 to 12. Intermediary XML files and extracted text with taxonomical hierarchy and page numbers have been released.

Additionally, we created a dataset of 731 hand-labeled textbook pages to test our method with object detection baselines, using an 80:10:10 train, validation, and test split. Pages were converted to 416x416 pixel JPEG images, and three augmentations (horizontal flip, vertical flip, and random crop) were applied which led to the final 1755 images with 1901 total objects.

## 5 Experiments

### 5.1 Transcript extraction

We evaluated the performance of *Wav2Vec2 Large 960h* (Baevski et al., 2020) trained on the Librispeech ASR dataset (Panayotov et al., 2015), with and without our language model (*Wav2Vec2* and *Wav2Vec2-LM*), using Word Error Rate (WER), Match Error Rate (MER), and Word Information Lost (WIL) metrics. We compared it to the DeepSpeech ASR method (Amodei et al., 2016), with *Wav2Vec2* outperforming DeepSpeech in speed and accuracy. Both models ran on CPU, reporting average run-time per video in seconds. Our language model's inclusion improved domain-specific word transcription and reduced error rates, as shown in Table 1.

<sup>2</sup>NCERT Textbooks Webpage

Table 1: Performance of transcription methods

Method	WER	MER	WIL	Time
Deepspeech	0.238	0.234	0.359	117.2
Wav2Vec2	0.16	0.158	0.253	25.9
Wav2Vec2-LM	0.121	0.120	0.192	25.9

## 5.2 Pre-requisite detection

### 5.2.1 Performance on VID-REQ dataset

Upon evaluation, three models emerge as the top-performing models on our released dataset of 2,797 prerequisite video pairs (*VID-REQ*). These models—Extra Trees (Geurts et al., 2006), LightGBM (LGBM) (Ke et al., 2017), and Random Forest classifiers with linear SVC feature selection (RF-SVC) (Breiman, 2001)—are assessed using 5-fold cross-validation, reporting mean accuracy, precision, recall, and F1-score as shown in Table 2. Hyperparameters for each model were fine-tuned via grid-search from Scikit Learn (Pedregosa et al., 2011). Extra Trees emerged as the best-performing model with an F1-score of 79.08%. Although both Extra Trees and Random Forest employ multiple decision trees, the difference in performance can be attributed to their responses to various feature characteristics. The unique splitting mechanism of Extra Trees, which involves more randomness, lends robustness when dealing with potentially noisy or complex data. This resilience to the inherent complexities of the feature set likely contributed to Extra Trees’ superior performance over the LGBM and RF-SVC classifiers in our study. We employed the F1-score as a reliable metric given its simultaneous consideration of both precision and recall. This is crucial from a learner’s perspective, as it is vital to prevent mislabeling non-prerequisite videos as prerequisites while accurately identifying all essential prerequisite videos. Moreover, the F1 metric effectively addresses the slight class imbalance present in the dataset.

Furthermore, we replicate the approach outlined in Gasparetti (2022) on our dataset as a baseline comparison. This technique utilizes aggregated *fast-text* word-embeddings input into SVC and RF classifiers to predict prerequisite dependencies between pairs of textual documents. As demonstrated in Table 2, our method surpasses the baseline in all metrics, with an F1-score exceeding by more than 10%.

### 5.2.2 Performance on AL-CPL dataset

We also compared our features with those of (Liang et al., 2018b, 2019). The dataset released in Wang et al. (2016) is the most widely used Wikipedia pre-requisite dataset, which covers *data mining*, *geometry*, *physics*, and *pre-calculus* subjects. The authors of Liang et al. (2018b, 2019) have pre-processed this data which is released as the AL-CPL dataset. We extract our features from this dataset and quote F1-score performance using 5 fold cross validation of the best performing model i.e., Random Forest with linear SVC feature selection in Table 3. We also compare the results of this model with those of Miaschi et al. (2019) who have used a multimodal architecture that uses LSTM and global features similar to Liang et al. (2018b, 2019) to predict pre-requisites. Both the above mentioned methods quote mean 5-fold cross validation results for the F1 metric. However, Miaschi et al. (2019) has showcased performance on *in-domain* and *cross-domain* prerequisite relationships separately, on 3 variants of their proposed architecture (*M1, M2, M3*). Therefore, in order to facilitate direct comparison we choose best results for the F1-score across the models and then take average of the *in-domain* and *cross-domain* results. As evident in Table 3 our method surpasses Liang et al. (2018b, 2019) for all subjects and Miaschi et al. (2019) for 3 out of 4 subjects. The average F1-score across subjects of our methods also surpasses that of Miaschi et al. (2019).

### 5.2.3 Performance on Meta-Academy dataset

We further showcase performance of our method on another Wikipedia pre-requisite dataset that includes pre-requisites extracted from Meta-Academy (Sayyadiharikandeh et al., 2019). Meta-academy is a free, open-source platform encompassing 487 machine-learning concepts connected by 7,947 prerequisite pairs. Our top-performing model, RF-SVC, trained on our novel features, demonstrates superior performance compared to the AdaBoost model trained on *Wiki-clicks-based* features (user navigation patterns on Wikipedia) on this dataset. As exhibited in Table 2, our model surpasses the AdaBoost model across all metrics, with an F1-score exceeding by over 5%.

These experiments showcase the robustness of our features, exceeding benchmarks for Wikipedia prerequisites tasks, even though they were designed for videos. This success can be attributed to our in-depth collaboration with domain expert teach-

Table 2: A comparative analysis of our prerequisite detection method and other methods across multiple datasets.

Dataset	Method	Model	Accuracy	Precision	Recall	F1-Score
VID-REQ (ours)	Gasparetti (2022)	RF	77.53	76.72	62.63	68.84
		SVC	75.11	69.22	67.79	68.44
	Ours	Extra Trees	<b>84.09*</b>	<b>82.85*</b>	75.83	<b>79.08*</b>
		LGBM	83.01	80.48	75.74	78.00
		RF(SVC)	83.12	79.82	<b>77.36*</b>	78.43
Meta Academy	Wiki-Clicks	Ada-Boost	81	80	78	80
	Ours	RF(SVC)	<b>84*</b>	<b>85*</b>	<b>85*</b>	<b>85*</b>

Table 3: F1-scores for various methods performed across different subjects on the AL-CPL dataset.

Dataset	Method	DataMining	Geometry	Physics	PreCalculus	Avg.
AL- CPL	Miaschi et al. (2019)	78.1	89.1	81.8	<b>91*</b>	85
	Liang et al. (2018a)	76.7	89.5	69.9	88.6	81.1
	Ours	<b>80.7*</b>	<b>90.4*</b>	<b>83*</b>	89.2	<b>85.8*</b>

ers during feature creation, leading to enhanced effectiveness and performance of our algorithm.

### 5.3 PDF Parsing

To evaluate performance on the dataset described in Section 4.3, we use an 80:20 train-test split. The LightGBM classifier (Ke et al., 2017) achieves the best classification results as shown Table 4 and is used in the PDF parser to generate our K-12 corpus.

Table 4: Performance of LGBM Classifier

Text class	Precision	Recall	F1
Chapter names	0.78	0.64	0.70
Section names	1.00	0.57	0.73
Text body	0.94	0.98	0.96
<b>Average</b>	<b>0.9067</b>	<b>0.73</b>	<b>0.7967</b>

To compare our PDF parsing methods with recent deep learning-based approaches, we treat the extraction of *text-classes* as an object detection problem, focusing on the crucial *section name* text class. We use a random subset of textbooks (46 section headers) and extract section headers using both methods. Headers are considered correctly matched if they have distance  $D$  (14) less than 0.6 (Doucet et al., 2011; Wu et al., 2013).

$$D = \frac{\text{LevenshteinDist}(A, B) * 10}{\text{Min}(\text{Len}(A), \text{Len}(B))} \quad (14)$$

For this experiment, we use the YOLOv5 model (Jocher, 2021) for object detection and EASYOCR (AI, 2021) to extract text from cropped header images. Our font-based classification method outper-

forms the YOLO + OCR approach in both performance and average per-page time as shown in Table 5. The deep learning method’s low precision stems from its reliance on visual features alone, which are inadequate for detecting *text-classes*. In contrast, our method utilizes text, color, and occurrence-based features for accurate classification, and by labeling only the fonts in PDF textbooks, it achieves faster and more precise performance.

Table 5: Comparison of our method with YOLO

Method	Preci-sion	Recall	F1 score	Time (in sec)
YOLO + OCR	0.533	0.869	0.661	2.54
Ours	<b>0.893</b>	<b>0.913</b>	<b>0.903</b>	<b>0.011</b>

## 6 Conclusion

In this paper, we present a pipeline for detecting prerequisite dependencies among academic videos using novel similarity-based features. Our approach outperforms existing methods, even surpassing prerequisite detection in domains like Wikipedia. We introduce hand-labeled datasets to discover prerequisite relations across diverse subjects, fostering future research in this area.

Future work will explore additional features and methods, extending our approach to a broader range of educational content such as podcasts, slides, and lecture notes. We also aim to integrate collaborative filtering and recommender systems for personalized learning paths, enhancing students’ educational experience and learning outcomes.

## References

- Jaided AI. 2021. Easyocr. <https://github.com/JaidedAI/EasyOCR>.
- Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. 2016. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182. PMLR.
- Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.
- Leo Breiman. 2001. Random forests. *Machine learning*, 45:5–32.
- Antoine Doucet, Gabriella Kazai, Bodin Dresevic, Aleksandar Uzelac, Bogdan Radakovic, and Nikola Todic. 2011. Setting up a competition framework for the evaluation of structure extraction from ocr-ed books. *International Journal on Document Analysis and Recognition (IJ DAR)*, 14(1):45–52.
- Fabio Gasparetti. 2022. Discovering prerequisite relations from educational documents through word embeddings. *Future Generation Computer Systems*, 127:31–41.
- Pierre Geurts, Damien Ernst, and Louis Wehenkel. 2006. Extremely randomized trees. *Machine learning*, 63:3–42.
- Jonathan Gordon, Linhong Zhu, Aram Galstyan, Prem Natarajan, and Gully Burns. 2016. Modeling concept dependencies in a scientific corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–875.
- Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197.
- Yilun Huang, Qinqin Yan, Yibo Li, Yifan Chen, Xiong Wang, Liangcai Gao, and Zhi Tang. 2019. A yolo-based table detection method. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 813–818. IEEE.
- Glenn Jocher. 2021. yolov5. <https://github.com/ultralytics/yolov5>.
- Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.
- R. Kneser and H. Ney. 1995. Improved backing-off for m-gram language modeling. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 181–184 vol.1.
- Lori J Leibold and Emily Buss. 2019. Masked speech recognition in school-age children. *Frontiers in Psychology*, 10:1981.
- Irene Li, Alexander R Fabbri, Robert R Tung, and Dragomir R Radev. 2019. What should i learn first: Introducing lecturebank for nlp education and prerequisite chain learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6674–6681.
- Chen Liang, Jianbo Ye, Shuting Wang, Bart Pursel, and C Lee Giles. 2018a. Investigating active learning for concept prerequisite learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- Chen Liang, Jianbo Ye, Han Zhao, Bart Pursel, and C Lee Giles. 2018b. Active learning of strict partial orders: A case study on concept prerequisite relations. *arXiv preprint arXiv:1801.06481*.
- Chen Liang, Jianbo Ye, Han Zhao, Bart Pursel, and C. Lee Giles. 2019. Active learning of strict partial orders: A case study on concept prerequisite relations. EDM 2019 - Proceedings of the 12th International Conference on Educational Data Mining, pages 348–353.
- Song Mao, Azriel Rosenfeld, and Tapas Kanungo. 2003. Document structure analysis algorithms: a literature survey. In *Document Recognition and Retrieval X*, volume 5010, pages 197–207. International Society for Optics and Photonics.
- Alessio Miaschi, Chiara Alzetta, Franco Alberto Cardillo, and Felice Dell’Orletta. 2019. Linguistically-driven strategy for concept prerequisites learning on italian. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 285–295.
- Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- John C Nesbit and Olusola O Adesope. 2006. Learning with concept and knowledge maps: A meta-analysis. *Review of educational research*, 76(3):413–448.

- Joseph D. Novak. 1990. [Concept mapping: A useful tool for science education](#). *Journal of Research in Science Teaching*, 27(10):937–949.
- Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.
- Yonggao Yang Kwang Paick Yanxiong Peng and Yukong Zhang. 2004. Pdf2xml: Converting pdf to xml.
- Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.
- Mohsen Sayyadharikandeh, Jonathan Gordon, Jose-Luis Ambite, and Kristina Lerman. 2019. Finding prerequisite relations using the wikipedia clickstream. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 1240–1247.
- Noah Siegel, Nicholas Lourie, Russell Power, and Waleed Ammar. 2018. Extracting scientific figures with distantly supervised neural networks. In *Proceedings of the 18th ACM/IEEE on joint conference on digital libraries*, pages 223–232.
- Mayank Singh, Barnopriyo Barua, Priyank Palod, Manvi Garg, Sidhartha Satapathy, Samuel Bushi, Kumar Ayush, Krishna Sai Rohith, Tulasi Gamidi, Pawan Goyal, et al. 2016. Ocr++: a robust framework for information extraction from scholarly articles. *arXiv preprint arXiv:1609.06423*.
- Christopher G Stahl, Steven R Young, Drahomira Herrmannova, Robert M Patton, and Jack C Wells. 2018. Deeppdf: A deep learning approach to extracting text from pdfs. Technical report, Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States).
- Suramya Tomar. 2006. Converting video formats with ffmpeg. *Linux Journal*, 2006(146):10.
- Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Pino. 2020. fairseq s2t: Fast speech-to-text modeling with fairseq. *arXiv preprint arXiv:2010.05171*.
- Shuting Wang, Alexander Ororbia, Zhaohui Wu, Kyle Williams, Chen Liang, Bart Pursel, and C Lee Giles. 2016. Using prerequisites to extract concept maps from textbooks. In *Proceedings of the 25th acm international on conference on information and knowledge management*, pages 317–326.
- Zhaohui Wu, Prasenjit Mitra, and C Lee Giles. 2013. Table of contents recognition and extraction for heterogeneous book documents. In *2013 12th international conference on document analysis and recognition*, pages 1205–1209. IEEE.

# Transformer-based Hebrew NLP models for Short Answer Scoring in Biology

Abigail Gurin Schleifer<sup>1</sup> Beata Beigman Klebanov<sup>2</sup> Moriah Ariely<sup>1</sup> Giora Alexandron<sup>1</sup>

<sup>1</sup> Weizmann Institute of Science, Rehovot, Israel

<sup>2</sup> Educational Testing Service, Princeton, USA

{abigail.gurin-schleifer, moriah.ariely, giora.alexandron}  
@weizmann.ac.il  
bbeigmanklebanov@ets.org

## Abstract

Pre-trained large language models (PLMs) are adaptable to a wide range of downstream tasks by fine-tuning their rich contextual embeddings to the task, often without requiring much task-specific data. In this paper, we explore the use of a recently developed Hebrew PLM – alephBERT – for automated short answer grading of high school biology items. We show that the alephBERT-based system outperforms a strong CNN-based baseline, and that it generalizes unexpectedly well in a zero-shot paradigm to items on an unseen topic that address the same underlying biological concepts, opening up the possibility of automatically assessing new items without item-specific fine-tuning.

## 1 Introduction

Advances in NLP offer transformative technology to support educational practice, including scoring of constructed (free text) responses in both holistic and analytic fashion. In particular, pre-trained large language models (PLMs) hold great promise for applications that require sophisticated context-rich analysis of student responses.

However, progress in PLMs and their applications in English outstrips that in other languages. Recent research in Hebrew NLP made available a new Hebrew PLM – alephBERT (Seker et al., 2022); while it has been shown to be effective for NLP tasks such as POS tagging and NER, its effectiveness for a downstream automated scoring application is an open question.

We evaluate alephBERT-based classifiers for the task of analytic content-scoring of short answers in biology in a formative high school setting, comparing it to a strong CNN-based baseline.

We contribute new knowledge about the effectiveness of BERT-based classifiers in languages other than English for a content-scoring task. Our two key findings are that the alephBERT-based classifiers i) provide a significant improvement over

the CNN-based baseline; and ii) generalize surprisingly well to unseen items that deal with the same underlying scientific concepts but in the context of a different topic. We briefly discuss implications of the findings and directions for future work.

## 2 Related Work

An especially promising application area of NLP is automated analysis of responses to open-ended questions, either in the form of a full essay, where the goal is typically a demonstration of proficiency in writing in a particular genre (Beigman Klebanov and Madnani, 2021), or in the form of short responses, where the goal is typically to demonstrate content knowledge. In this paper, we consider the latter application, often termed Automated Short Answer Grading (ASAG).

To date, most of the scientific development on ASAG has been done in English (see Haller et al. (2022) for a survey), including ASAG using PLMs (Bexte et al., 2022; Li et al., 2021; Condor, 2020; Sung et al., 2019a,b), although work on PLMs for ASAG in other languages does exist, e.g., Japanese (Oka et al., 2022), Arabic (Nael et al., 2022).

Recently researchers also used multi-lingual PLMs for ASAG: Schneider et al. (2023) used the LaBSE multilingual transformer model (Feng et al., 2022) for scoring very short responses (the bulk of the responses are 5 words or shorter) in a variety of subjects and in 14 languages. Unfortunately, the authors did not provide a detailed breakdown of performance by language or by subject area, although they did show that numeric responses tended to be easier to score than textual or mixed ones, across multiple languages. Interestingly, while there were relatively few responses in English (1.7K), the system’s error on scoring textual responses in English was lower than for Ukrainian, which had more than two orders of magnitude more responses than English (500K), which could suggest that languages with smaller digital footprints and therefore less

data for pre-training the PLMs would still be at a disadvantage even if there are a lot of responses in those languages for the specific task.

The ASAG task for Hebrew was addressed by Ariely et al. (2023). The authors built CNN-based classifiers that used word2vec embeddings; these models will serve as baselines for the current work. Hebrew, like Arabic, is a semitic language where vowels are generally omitted in writing, resulting in substantial ambiguity where the same sequence of written letters can have many meanings depending on context. Therefore, a PLM that implements the latest contextualization advancements holds great promise for ASAG in Hebrew. AlephBERT, the recently introduced Hebrew PLM (Seker et al., 2022), shows SOTA performance on multiple tasks, including morphological and POS tagging and NER. Our goal is to evaluate alephBERT for the ASAG task in Hebrew.

### 3 Experimental Setup

#### 3.1 Data

The data consists of responses to open-ended questions on three biology items from 669 students in grades 10-12 from about 25 high schools across Israel. There are thus 669 labeled responses for each of the three items (henceforth, **q1**, **q2**, **q3**), scored by a team of content and pedagogy experts with a binary score per category; that is, for every response, there are 10-13 binary labels according to the analytic rubric for the given item.

The items present questions about the effect of smoking (q1), anemia (q2), and travel in high altitude (q3) on physical activity. A very similar analytic rubric is used for all three items to assess students' ability to write causal explanations in biology. The rubric consists of a causal reasoning chain built from 13 categories, each of which evaluates whether a specific scientific fact or causal relation is addressed correctly in a response. Table 1 shows the mapping between the items and the binary analytic categories. Table 2 shows brief definitions of the categories. Figure 1 shows the score distributions per item per category. We observe that item q3 is harder than items q1 and q2 on most categories shared by the three items.

The rubric evaluates the ability to explain step-by-step the causal chain leading to the phenomenon. For example, q1 asks students to explain how high levels of CO make it difficult for smokers to exercise. Two responses are shown below, trans-

Item	Categories
q1	-,1,-,3,4,5,6,7,8,9,10,11,12
q2	-,,-,3,4,5,6,7,8,9,10,11,12
q3	0,1,2,3,4,5,6,7,8,9,10,11,12

Table 1: The mapping between items and categories.

Cat	Definition
0	changes in the amount of RBC
1	changes in oxygen levels that bind to HGB/RBC
2	refer to both groups of athlete travelers (q3)
3	the role of HGB/RBC in oxygen transportation
4	changes in oxygen levels in the body
5	changes in oxygen levels in the cells
6	oxygen is a reactant in cellular respiration
7	energy/ATP is produced during cellular resp.
8	changes in cellular respiration rate
9	using the term 'cellular respiration'
10	changes in energy/ATP levels
11	using the term 'energy' or 'ATP'
12	energy is consumed during exercise

Table 2: Category definitions. HGB: hemoglobin; RBC: red blood cells; ATP: energy (adenosine triphosphate ).

lated into English. Response 1 was given credit for mentioning the changes in oxygen levels after CO binding to hemoglobin (category 1), for stating the connection between the decreased cellular respiration rates and the reduction in the generation of energy which is necessary for physical activity (8-12). However, the reasoning chain is not articulated fully, since the transfer of oxygen to the cells by red blood cells and the role of oxygen in cellular respiration are not stated (no credit for categories 3-7). Conversely, Response 2 does mention the impairment of oxygen transfer to the body and cells (4 and 5), but does not include the parts of the explanation that connect oxygen to cellular respiration and cellular respiration to production of energy for the physical activity, hence no credit is given on categories 6-12.

**Response 1** A cigarette contains several harmful substances, including CO. CO has a strong tendency to bind to hemoglobin found in red blood cells. As a result, less oxygen binds to hemoglobin, which affects the rate of cellular respiration. Because the rate of cellular respiration slows down, less energy is generated in the cells of the body, so the cells do not have enough energy to perform physical activity and difficulty is created. Scores: [-, 1, -, 0, 0, 0, 0, 1, 1, 1, 1, 1]

**Response 2** Because those carbon dioxide molecules bind to hemoglobin, the transfer of oxygen to the body's cells is impaired.

Lack of hemoglobin and oxygen explains the difficulty of people who smoke to exercise.  
 Scores: [-, 1, -,0,1,1,0,0,0,0,0,0,0]

This rubric was developed in consultation with teachers to support in-class formative assessment, for example by assigning students to small study groups based on reasoning types revealed in their response patterns.

The items are typical open-ended questions commonly used (or versions of them) in teaching materials in biology and in the Israeli high school matriculation exam ('Bagrut'). The three items were presented to students in a randomized order. The average length of response is 55, 48, 70 words and standard deviation of 34.5, 27.4, 48 for q1, q2, q3, respectively. The data collection was approved by IRB and includes permission to use the data for research. The data was collected prior to and independently of this study and was previously used in computational experiments of Ariely et al. (2023).

### 3.2 Experiment design

In this study, we investigate how well an alephBERT classifier performs on analytic ASAG, compared to the CNN-based system of Ariely et al. (2023). We conduct evaluations in two scenarios: (a) within-item, where train and test data come from the same item, and (b) cross-items, where the system is trained on two items and tested on the third. The main goal of the latter evaluation is to address cases where a new item is created that deals with a different application area of the same scientific concept, that is, a new item that would address cellular respiration mechanism in a different real-life application. This is a common pedagogical strategy for creating teaching, practice, and multiple forms of assessment materials.

We partition the students into train, development, and test groups in the 60/20/20 proportions respectively; their responses comprise the q1-train, q1-dev, q1-test sets, and the same for q2 and q3. This is done in order to ensure that responses from the same student do not appear in both train and test data in the evaluations. We build a classifier for each category (13 classifiers in total); while the student responses are the same across categories (we are using the full text of the response), the labels may differ across categories. That is, a given response can have the score of 0 on category 3 and the score of 1 on category 8, as in Response 1 shown in section 3.1.

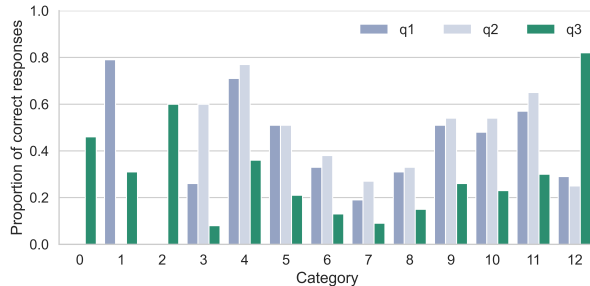


Figure 1: Proportion of correct responses per item per category.

For within-item experiments, we train on q1-train and test on q1-test; same for q2 and q3. For cross-item experiments, we train on the combination of q1-train and q2-train and test on q3-test; the same for the other two permutations of the items. In this design, in addition to benchmarking against prior work, we also compare performance between within-item and cross-item scenarios, e.g., results on q3-test when trained on q3-train vs trained on the combination of q1-train and q2-train.

For evaluation, we use Cohen’s  $\kappa$ , per item per category. We also report proportion of categories with  $\kappa > 0.6$ , to get a sense of the extent to which the rubric as a whole can be automatically scored with reasonable reliability for a formative context. Ariely et al. (2023) reported average performance over 50 iterations of cross-validation for each item and each category; in our context, it is prohibitively time-consuming to run such a large number of evaluations. We report evaluations on q1, q2, and q3 test sets for the alephBERT models; thus, performance estimates for alephBERT are somewhat noisier than for the CNN baseline.

## 4 Models

### 4.1 Baseline

For the baseline, we use published results for CNN-based classifiers reported in Ariely et al. (2023), where each classifier predicts whether a certain category is addressed in the response. Pre-processing included tokenizing the input text and performing a morphological and syntactic analysis using Hebrew NLP tools. Word embeddings over a vocabulary of frequently-used morphemes and their part of speech were constructed using Gensim’s word2vec CBOW algorithm. The embeddings were fed forward into two consecutive convolutional layers, followed by a fully connected layer and a sigmoid activation function. The embeddings (of size 100)



were trained on the entire Hebrew Wikipedia.

## 4.2 AlephBERT based models

AlephBERT PLM (Seker et al., 2022) is based on the same architecture as the English BERT PLM (Devlin et al., 2018). AlephBERT was designed to handle Hebrew morphology; see Seker et al. (2022) for a detailed description. AlephBERT was trained on a larger corpus than any Hebrew language model before it, including Twitter, Hebrew wiki and the Hebrew portion of the Oscar dataset (Ortiz Suárez et al., 2020). It was not specifically trained on biology or science data beyond the occurrence of these topics in the general corpora. It includes 12 layers, i.e., transformer blocks (768 units per layer), 12 attention heads, the total of 110M parameters and vocabulary size of 52K.

For every category, we built a classifier that uses the alephBERT PLM pre-trained embeddings and an additional classification layer, with sigmoid activation. We fine-tune the models on our training data using cross-entropy loss; all layers of the model are tuned. The learning rate and number of epochs hyperparameters were tuned on dev sets.

## 5 Results

Table 3 shows the performance of the alephBERT-based system on all <category, item, case> combinations, where case refers to ‘within-item’ or ‘cross-item’. The performance of the CNN baseline is shown as published in Ariely et al. (2023).

### 5.1 Comparison to CNN baseline

AlephBERT-based models perform significantly better than the baseline,  $p = 0.016$ , using the one-sided Wilcoxon signed-rank test (paired) with  $n = 44$  (all <item,category,case> cells in Table 3 that have results for both the models),  $\alpha = 0.05$ . The largest gain is on category 9 within-item: from  $\kappa = .06-.73$  (baseline) to  $\kappa > .90$  (alephBERT). Category 9 looks for a specific phrase (‘cellular respiration’). We hypothesize that this improvement is driven by the improved ability of alephBERT to capture the rich token-internal structure of the Hebrew language reported by Seker et al. (2022) based on morpheme-level evaluations.

### 5.2 Comparison between within-item and cross-item performance

We compare the alephBERT-based within-item models with the cross-items (i.e., zero-shot) models

on all <category, item> combinations where both models can be run (see Table 3). The cross-item performance is not significantly worse than within-item,  $p = 0.9$  using the one-sided Wilcoxon signed-rank test (paired),  $n = 32$ ,  $\alpha = 0.05$ .

This is a remarkable result, since one would expect a degradation in performance for models that saw no data coming from the test item at train time. In fact, an unseen item on the same biology concept can be scored with a common analytic rubric with  $\kappa > 0.6$  on average across categories for each item, which may be sufficient for formative uses and may allow teachers to create and score new items based on a similar rubric on the fly.

We observe a complete failure of cross-item generalization on category 1. This category occurs only in q1 and q3; the cross-item generalization is thus based on one training item. This could compromise the system’s ability to zero in on those meaning elements that are common to the two training items and instead overly rely on the specifics of the training item’s topic. Category 1 is also more difficult to address well in q3 than in q1 (30% correct vs 78% correct, see Figure 1), further complicating cross-item transfer. Understanding the necessary conditions for transfer is a topic for future research.

## 6 Conclusions

Pre-trained large language models can be adapted to downstream tasks by fine-tuning their rich contextual embeddings to the task. We explored the recent Hebrew PLM – alephBERT – for short answer grading in high school biology. We found that the alephBERT-based system outperformed a strong baseline and that it generalized unexpectedly well to items on an unseen topic addressing the same biology concepts. The second finding provides evidence in support of the viability of the modular design of the rubric – not only is it the case that human raters were able to reliably assess different items with subsets of the same analytic categories, but an automated model was likewise able to zero in on the commonalities in the way categories are manifested in student responses across multiple topics.

The cross-item generalization has exciting implications for educational practice, as this may allow teachers to create and automatically score new items based on a similar rubric on the fly. A study of this possibility with teachers and an improvement of our understanding of the conditions neces-

Category↓	Model→	Ariely2023				AlephBERT							
	Item→	q1		q2		q3		q1		q2		q3	
	Case→	W-I	W-I	W-I	C-I	W-I	C-I	W-I	C-I	W-I	C-I	W-I	C-I
0				.71								.81	
1		.53		.76		.72	.00					.61	.01
2				.70								.88	
3		.60	.73	.00	.48	.75	.43	.62	.54	.00	.67		
4		.61	.52	.60	.38	.50	.61	.35	.05	.71	.47		
5		.80	.75	.57	.76	.90	.76	.73	.66	.81	.79		
6		.66	.72	.32	.71	.65	.69	.71	.68	.66	.59		
7		.71	.80	.47	.61	.68	.51	.73	.78	.50	.76		
8		.95	.93	.93	.70	.85	.86	.93	.72	.32	.82		
9		.46	.73	.06	.95	.99	.97	.96	.97	.94	.96		
10		.83	.80	.60	.80	.88	.88	.65	.71	.97	.87		
11		.91	.90	.90	.93	.97	.97	.88	.87	.95	.95		
12		.68	.57	.00	.61	.74	.00	.73	.75	.00	.54		
Av		.70	.75	.51	.69	.78	.61	.73	.67	.63	.68		
% $\kappa > .60$		73	80	38	80	91	64	90	80	69	64		

Table 3: Average Cohen’s  $\kappa$  per item (q1-q3) per category (0-12), for the baseline as reported in Ariely et al. (2023) and alephBERT models. W-I: within-item (gray); C-I: cross-items. The last row shows % of categories with  $\kappa > 0.6$ .

sary for the successful transfer to occur are two of the directions of our future work, as well as further enhancement of the scoring system.

## Acknowledgements

This research was partially supported by the Israeli Council for Higher Education (CHE) via the Weizmann Data Science Research Center.

## References

- Moriah Ariely, Tanya Nazaretsky, and Giora Alexandron. 2023. Machine learning and Hebrew NLP for automated assessment of open-ended questions in biology. *International Journal of Artificial Intelligence in Education*, 33(1):1–34.
- Beata Beigman Klebanov and Nitin Madnani. 2021. Automated essay scoring. *Synthesis Lectures on Human Language Technologies*, 14(5):1–314.
- Marie Bexte, Andrea Horbach, and Torsten Zesch. 2022. Similarity-based content scoring-how to make s-bert keep up with bert. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 118–123.
- Aubrey Condor. 2020. Exploring automatic short answer grading as a tool to assist in human rating. In *Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part II 21*, pages 74–79. Springer.
- 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*.
- Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. [Language-agnostic BERT sentence embedding](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.
- Stefan Haller, Adina Aldea, Christin Seifert, and Nicola Strisciuglio. 2022. Survey on automated short answer grading with deep learning: From word embeddings to transformers. *arXiv preprint arXiv:2204.03503*.
- Zhaohui Li, Yajur Tomar, and Rebecca J. Passonneau. 2021. [A semantic feature-wise transformation relation network for automatic short answer grading](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6030–6040, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Omar Nael, Youssef ELmanyawaly, and Nada Sharaf. 2022. AraScore: A deep learning-based system for Arabic short answer scoring. *Array*, 13:100109.
- Haruki Oka, Hung Tuan Nguyen, Cuong Tuan Nguyen, Masaki Nakagawa, and Tsunenori Ishioka. 2022. Fully automated short answer scoring of the trial tests for common entrance examinations for japanese university. In *Artificial Intelligence in Education:*

23rd International Conference, AIED 2022, Durham, UK, July 27–31, 2022, *Proceedings, Part I*, pages 180–192. Springer.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. [A monolingual approach to contextualized word embeddings for mid-resource languages](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics.

Johannes Schneider, Robin Richner, and Micha Riser. 2023. Towards trustworthy autograding of short, multi-lingual, multi-type answers. *International Journal of Artificial Intelligence in Education*, 33(1):88–118.

Amit Seker, Elron Bandel, Dan Bareket, Idan Brusilovsky, Refael Greenfeld, and Reut Tsarfaty. 2022. [AlephBERT: Language model pre-training and evaluation from sub-word to sentence level](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 46–56, Dublin, Ireland. Association for Computational Linguistics.

Chul Sung, Tejas Dhamecha, Swarnadeep Saha, Tengfei Ma, Vinay Reddy, and Rishi Arora. 2019a. Pre-training bert on domain resources for short answer grading. 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 6071–6075.

Chul Sung, Tejas Indulal Dhamecha, and Nirmal Mukhi. 2019b. Improving short answer grading using transformer-based pre-training. In *Artificial Intelligence in Education: 20th International Conference, AIED 2019, Chicago, IL, USA, June 25-29, 2019, Proceedings, Part I 20*, pages 469–481. Springer.

# Comparing Neural Question Generation Architectures for Reading Comprehension

E. Margaret Perkoff and Abhidip Bhattacharyya and Jon Z. Cai and Jie Cao

University of Colorado Boulder

firstname.lastname@colorado.edu

## Abstract

In recent decades, there has been a significant push to leverage technology to aid both teachers and students in the classroom. Language processing advancements have been harnessed to provide better tutoring services, automated feedback to teachers, improved peer-to-peer feedback mechanisms, and measures of student comprehension for reading. Automated question generation systems have the potential to significantly reduce teachers' workload in the latter. In this paper, we compare three different neural architectures for question generation across two types of reading material: narratives and textbooks. For each architecture, we explore the benefits of including question attributes in the input representation. Our models show that a T5 architecture has the best overall performance, with a RougeL score of 0.536 on a narrative corpus and 0.316 on a textbook corpus. We break down the results by attribute and discover that the attribute can improve the quality of some types of generated questions, including *Action* and *Character*, but this is not true for all models.

## 1 Introduction

The task of Automated Question Generation (AQG) has been proven to have significant potential for reducing teacher workload while effectively assessing reading comprehension for students (Kurdi et al., 2020). Reading comprehension is indicative of a student's understanding of a subject, making it a critical metric for ensuring their future academic success. Originally, advancements in Question Generation were isolated to broad question answering datasets including SQuAD (Rajpurkar et al., 2016) and NarrativeQA (Kočíský et al., 2018). In recent years, AQG models pre-trained on these datasets have been applied to education-specific corpora to help generate questions that are more useful in the classroom setting.

The domain shift from generic corpora to education-specific datasets is critical to model the

unique characteristics of classroom discourse, but there is still much room for improvement. Classroom texts vary greatly in terms of the age of the students, the subject material, and their discourse structure. Prior work in question generation for education has focused on how language models perform on a single corpus with a single subject (Xu et al., 2022a), but not on how these models perform across different subjects. This research also considers how different discourse representations for a particular neural architecture can improve the quality of generated questions as opposed to evaluating multiple systems. Here, we analyze how different neural models perform on two corpora: the FairytaleQA Corpus (Xu et al., 2022b) and the Textbook Question Answering (Kembhavi et al., 2017) dataset. The FairytaleQA Corpus is representative of narrative comprehension, whereas the Textbook Question Answering dataset focuses on scientific topics including Physical, Earth, and Life Sciences. In addition to covering different subjects these datasets are quite different in terms of the passage structure. Earlier research on AQG has also considered how different forms of discourse representation, such as question type and event summarization, can improve the quality of questions generated (Zhao et al., 2022; Zhou et al., 2019). The FairytaleQA corpus distinguishes questions by seven attribute types. These attributes indicate the semantic nature of the question as well as the type of information that the reader is searching for either implicitly or explicitly from the text. We incorporate the question attribute into each of our model architectures to see whether the attribute has more significant impact when combined with a particular neural structure.

In this paper, we compare the performance of three different neural AQG architectures across two different datasets. We train baseline models for AQG on the FairytaleQA Corpus, (Xu et al., 2022b) a narrative dataset for K-12 reading com-

prehension, and the Textbook Question Answering (Kembhavi et al., 2017) dataset focused on middle school science. These models include a T5 (Raffel et al., 2019), BART (Lewis et al., 2019), and GPT-2 (Radford et al., 2019). We also investigate the impact of incorporating question attributes into these different model types for the FairytaleQA corpus. The T5 models achieve the highest metric rankings across both datasets, with BART outperforming GPT-2 on both as well. Including the question attribute as part of the input for training and inference improves the overall results for all model types, but leads to greater performance improvements for the GPT-2 and T5 models than for the BART model. Additionally, our by attribute breakdown found that including question attribute does not increase ROUGE scores for setting attribute questions. To our knowledge, this is the first comparison of a broader set of neural architectures for AQG in the education domain. These baselines are intended to inform future work on AQG in the classroom while taking into account the nuances of different subjects.

## 2 Related Works

### 2.1 Question Generation for the Education Domain

Significant amount of prior work addressed automated question generation (AQG) methods in the classroom. A review by Kurdi et al. (2020) concluded that AQG had the potential to provide significant benefit to teachers and students. Teachers can leverage question generation methods to automate assessment creation and reduce their workload. Question generation can also benefit students when used in tutoring or student-led learning contexts. Wang et al. (2018) introduced QG-Net, the earliest application of a model pretrained on a more general dataset (in this case SQuAD (Rajpurkar et al., 2016)) to the classroom material. They fine-tuned their model on the OpenStax textbooks<sup>1</sup>. The work of Zou et al. proposed an unsupervised method to generate true / false questions for reading comprehension. They compared a template-based framework and a pretrained BART model for text infilling. In human evaluations, the framework models outperformed the generative model in all categories except Fluency.

In 2022, Xu et al. (2022a) introduced the FairytaleQA dataset that we use for fine-tuning and es-

<sup>1</sup><https://openstax.org/k12>

tablish a baseline for generating questions with a fine-tuned BART (Lewis et al., 2019) question answering model. They discovered that fine-tuning on the FairytaleQA dataset outperforms the BARTQA model fine-tuned on the NarrativeQA and FairytaleQA (Kočíský et al., 2018) corpora. Additionally, the distribution of attributes of the generated questions more closely resembled the distribution of the questions generated by expert human annotators. Their work implied the importance of fine-tuning models on domain specific datasets with high quality questions for reading comprehension. Rathod et al. introduced the concept of Multi Question Generation in the educational domain to create more lexically diverse questions that have the same answer.

This prior work in the education space has focused experiments largely on a single model architecture - BART, but has not considered more recent improvements in neural generative architectures. Grover et al. (2021) explored the use of a pre-trained T5 transformer model for the task of question generation without answer supervision. Their model was designed to take a passage as input and output multiple question-answer pairs related to the passage. It was trained and evaluated on the SQuAD dataset (Rajpurkar et al., 2016) for general question answering, but was not applied to education specific datasets. Based on their results, we evaluate the effectiveness of T5 in the education domain in our experiments.

Laban et al. (2022) looked beyond just generating quiz questions and conducted an experiment to evaluate generation errors. The result questions are categorized define a hierarchy of errors with three top-level justifications: *disfluent*, *off target*, and *wrong context*. Included among their models are three GPT-2 based models as well as two BART models - all of which are fine-tuned on the SQuAD dataset. The BART-large model has the second lowest rate of errors under their system with the GPT-2 based models all performing at the lower end of the range. Their experiment setup for both BART and GPT-2 does not fine-tune on pedagogical texts, so we will be able to explore if this boosts performance in our experiments.

### 2.2 Question Generation With Question Type or Attribute

Researchers have explored the use of question types or attributes to enhance question generation both

Dataset	Train	Valid	Test
FairytaleQA	6000	504	485
Textbook Question Answering	3346	1029	1074

Table 1: Breakdown of the datasets by training, validation, and test splits. Each sample includes an answer, a gold question, and section text from the relevant reading.

within and outside of a learning context. Zhou et al. proposed a model that would jointly predict the question type and generate a question. They distinguish between 8 types - seven types for different question words (*what, who, when, why, how, which, where*) and one *others* category. Their unified model outperformed earlier AQG methods on both the SQuAD and MARCO (Nguyen et al., 2016) datasets. Wang et al. sought to improve the diversity of generated questions by leveraging a conditional variational auto-encoder (CVAE) that incorporates the question types proposed in (Zhou et al., 2019). The CVAE approach demonstrated that incorporating question type did improve diversity of responses both on SQuAD and NewsQA (Trischler et al., 2017). Zhao et al. applied the idea of question type informed AQG to the FairytaleQA corpus. However, their approach involves taking a story passage as input and predicting the distribution of question types (noted as attributes in the context of the FairytaleQA data) to inform question generation. This distribution is then fed to an event-centric summary generation model and ultimately that output is passed on to a BART-based question generation model. Most of the aforementioned models were built with a pre-trained BART backbone, and none of these approaches considered using a T5 or GPT-model for the generation step. In our experiments, we incorporate the attribute value from the FairytaleQA corpus into all three of these model variants to compare the impact across different architectures.

### 3 Experimental Set Up

#### 3.1 Datasets

We used two datasets for our experiments: Fairytale QA Corpus and Textbook Question Answering (TQA). A brief summary of the datasets is presented in Table 1. For the AQG task, we required having our data in the format of a story passage, or context  $C$ , an anticipated answer  $a$ , and a gold

FairytaleQA
<b>story:</b> It so happened that Finn and his gigantic relatives were all working at the Giant’s Causeway in order to make a bridge, ... <b>question:</b> Why were Finn and his gigantic relatives at the Giant’s Causeway? <b>answer:</b> to make a bridge <b>attribute:</b> causal relation
TQA
<b>context:</b> A cold front occurs when a cold air mass runs into a warm air mass. This is shown in Figure 16.7. The cold air mass moves faster than the warm air mass and lifts the warm air mass out of its way. As the warm air rises, its water vapor condenses ... <b>question:</b> A warm front occurs when <b>answer:</b> a warm air mass slides over a cold air mass

Table 2: Question-Answer pair examples from FairytaleQA and TQA dataset

standard question  $g$ . During training, the goal is to generate a question that is as close (syntactically and semantically) to the gold question.

#### 3.1.1 Fairytale QA Corpus

We use the FairytaleQA Corpus (Xu et al., 2022b) to assess the ability of our models to create meaningful questions based on narratives. This corpus contains 10,580 question-answer pairs based on 278 children-friendly stories. These pairs were created by annotators with expertise in education, cognitive science, and/or psychology. Each pair is labelled with the relevant story section. The corpus is further broken down into seven types of attributes: *character, setting, action, feeling, causal relationship, outcome resolution, and prediction*. All questions are also annotated as explicit or implicit - based on whether or not the answer to the question is explicitly stated in the corresponding text passage. Table 2 depicts an example from the fairytale dataset.

#### 3.1.2 Textbook Question Answering

The Textbook Question Answering (TQA) dataset (Kembhavi et al., 2017) is based on questions from middle school textbooks in life science, earth science, and physical science. The original version contains 26,260 questions that can be used to train models for text-based and visual question answering. It is structured such that questions are associated with a particular lesson, but not the text passage from which the answer is drawn. Each lesson contains a set of topics along with a description of topic content. The questions are in a multiple-choice format and includes questions that refer to figures that are present in the text. We go

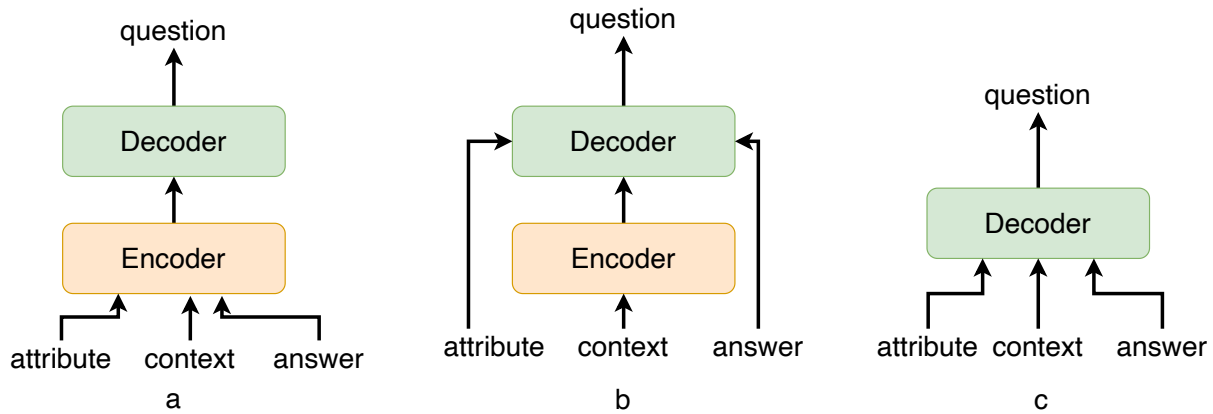


Figure 1: Architecture comparison: (a) represents T5 architecture, where the encoder takes attribute, context and answer text all together as input. (b) represents BART architecture, where encoder takes context as input and decoder takes attribute and answer as generation prefix. (c) represents GPT-2 architecture, where the decoder takes attribute, context and answer as generation prefix.

through the following preprocessing steps to make the dataset work for our text-based AQG task.

1. Remove any question that refers to a diagram.
2. Remove any question-answer pairs that require knowledge of more than one of the answer options, such as *Which of the following is false*, *None of the above* or *Answers B and C*.
3. For the remaining questions:
  - (a) Extract the text from the correct answer label to use as the answer
  - (b) Select the text passage or passages (in the case of a tie) with the highest word overlap between the passage and the question and answer to use as the context

The resulting dataset contains 3,346 question-answer pairs with context for training. Table 2 depicts an example from TQA dataset.

## 3.2 Models

We use three different pre-trained language models as our base model to further fine-tune on FairyTaleQA and TQA datasets to test the impact of different architectures and pre-training objectives to question generation task.

### 3.2.1 T5 Models

We use the T5 base model (Raffel et al., 2019) available from the huggingface library as the first example of a sequence-to-sequence architecture. T5 models treat all tasks as a text-to-text format, where

the encoder takes source sequence as input and the decoder learns the generate output sequence. For question generation, the input text includes at minimum the question task indicator, a context passage, and then outputs a question. The encoder-decoder architecture can be seen in 1 a. The model is fine-tuned separately on each dataset for a total of 10 epochs with a learning rate of  $1e^{-4}$ . The attribute-based model for the FairyTaleQA dataset includes the attribute along with a special token `attribute:`.

### 3.2.2 BART Models

Our second model is another encoder-decoder model. We use the BART base model (Lewis et al., 2019). To be specific, we deployed BartForConditionalGeneration from the huggingface library. Unlike the T5 model, we provide only the context text as input to the encoder. The attribute and answer was given to the decoder. The motivation was to enable the encoder to create a holistic representation of the context which can further be queried by the decoder with specific information. We trained the model for 50 epochs to learn both question generation and answer generation. During this training period for each data, with 50% probability the mode will be switched to either question generation or answer generation. During question generation the decoder will have the ground truth attribute and the answer. For answer generation, the decoder will have the ground truth attribute and the question. We further fine-tuned the model for 10 more epochs for question generation.

### 3.2.3 GPT-2 Models

The third model is a pre-trained GPT-2 model that leverages a pure decoder architecture (Radford et al., 2019). GPT-2(GPT-2 base model, 117M parameters) was trained on large amount of text with left-to-right Language Modeling objective, namely modeling the joint probability of a sequence of tokens in a left-to-right fashion of decomposition. The simplistic pre-training paradigm has been adopted by bigger and more powerful model successors such as GPT3, GPT4 and Llama(Brown et al., 2020; Touvron et al., 2023). We choose to test the viability of encoding Question generation task with GPT-2 given the amount of resources available and cost. We fine-tuned the GPT-2 model for the question generation task encoded with a prompt. See Sec.4.2 for more details about how we encode the question generation task for GPT-2.

## 4 Experiments

### 4.1 Evaluation

We evaluate our results based on both automated metrics and qualitative analysis. To compare our results with those of previous work, we use two standard evaluation metrics: BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004). BLEU measures the similarity between the generated sentence and one or more reference sentences based on n-gram overlaps. ROUGE also considers n-gram overlaps but is a recall-focused measure, while BLEU is precision-focused. ROUGE gives more weight to n-gram matches that occur in multiple references. In the context of response generation, this means that if multiple candidate responses include a particular phrase, it will have a greater impact on the scoring of a specific response. While these metrics are useful for comparison purposes, they have been shown to have limited correlation with human judgments (Liu et al., 2016). In future work, we intend to evaluate responses with a group of human annotators with teaching and reading comprehension expertise. As part of our initial human evaluation, we have included a qualitative analysis to break down some of the responses generated in each domain.

### 4.2 Prompting

Each of the models uses different input prompts as visualized in Table 3. Prompting for a specific model was inspired by the model’s pre-training task. The T5 models are initially trained for

multiple text-to-text tasks so they require a special token for the task, an input text value, and an output for each training example. To fine-tune on our datasets, we use the special token `ask_question` for the task, the input text includes a special token for `answer`: followed by the answer and a special token for the `context`: including the relevant section text. The output text value is the anticipated question output followed by a special token to signal the end of the output `</s>`. In the BART model, the encoder was fed with a special task token  $\tau \in \text{both}, \text{ask\_question}$  and the story as the context. The beginning of the context is marked with a special token `context`. The target prompt consisted of mainly two elements- i) question-delineated by special tokens `<q>` and `</q>` and ii)answer-delineated by special tokens `<a>` and `</a>`. During the training for  $\tau$  as *both* the order of question and answer was changed with a probability of 50% to help the encoder to capture task agnostic information. During training under the task *ask\_question*, question was the last element in the target prompt. Note as per the design of BART, the decoder during training will have the target string right shifted by one position. During inference the decoder will have the question and it is expected to generate the correct target string with the question. In GPT-2 we encode the story context, answer, question and optionally attributes in natural language format as “`story section: {story_context} Now given an answer: {answer_text}, a good question would be {question_text}`” (without attribute), in which the placeholder variables within the curly parenthesis are filled with each story QA triplet. Table 3 depicts an example of input and output for each model. The vocabulary of GPT-2 differs from BART and T5 in terms of the special tokens that it contains only a end-of-sentence token in the existing vocabulary. We therefore follow the default vocabulary configuration and not include extra untrained tokens such as start-of-sentence and segmentation tokens.

### 4.3 Models with Attribute Input

For the experimental condition where we include the question attribute as part of the input, we modify the prompt for each model accordingly. We add an additional special token `attribute`:



<b>Training</b>	
T5	input_text: ask_question: answer: on Knockmany Hill context: Finn lived at this time on Knockmany Hill,... output_text: Where did Finn live </s>
BART	encoder: ask_question:context: Finn lived at this time on Knockmany Hill,... target: <s>attribute:<a>on Knockmany Hill</a><q>Where did Finn live</q></s> decoder: </s><s>attribute:<a>on Knockmany Hill</a><q>Where did Finn live</q>
GPT	story section: Finn lived at this time on Knockmany Hill,... Now given an answer: on Knockmany Hill and it is related to {attributes_text}, a good question would be Where did Finn live
<b>Inference</b>	
T5	input_text: ask_question: answer: on Knockmany Hill context: Finn lived at this time on Knockmany Hill... output_text: Where did Finn live</s>
BART	encoder: ask_question:context: Finn lived at this time on Knockmany Hill,... target: Where did Finn live</q></s> decoder: attribute:<a>on Knockmany Hill</a><q>
GPT	story section: Finn lived at this time on Knockmany Hill,... Now given an answer: on Knockmany Hill and it is related to {attributes_text}, a good question would be

Table 3: Comparison of training and inference prompt styles for the T5, BART, and GPT models. The gold standard question from the dataset is: "Where did Finn live?" and the gold answer is "on Knockmany Hill". The full context of the story includes mention to the main character, a giant named Finn, his wife Oonagh, and his gigantic relations who reside on Knockmany Hill in Ireland. For brevity in the examples, we do not include the entire passage in the prompt table above.

to the input text for the T5 model, which is then followed by the corresponding question attribute for each training sample. At inference time, the attribute is also included as part of the input, and there is no change to the output text values for training or inference. For BART, the output prompt is modified by prepending the specific attribute token. For GPT-2, the prompt is modified as "story section: {story\_context} Now given an answer: {answer\_text} and it is related to {attributes\_text}, a good question would be {question\_text}", with all attributes concatenated with comma within the attributes\_text variable.

## 5 Results

The result BLEU and RougeL scores across both datasets can be seen in Table 4. We found that the T5 models outperform all of the BART and GPT variations on both datasets. Our BART architecture

Model	Dataset	RougeL	BLEU
T5	FairytaleQA	<b>0.536</b>	<b>0.307</b>
T5-attr	FairytaleQA	0.500	0.279
BART	FairytaleQA	0.372	0.175
BART-attr	FairytaleQA	0.372	0.191
GPT	FairytaleQA	0.281	0.086
GPT-attr	FairytaleQA	0.295	0.087
T5	TQA	<b>0.316</b>	<b>0.107</b>
BART	TQA	0.166	0.042
GPT	TQA	0.089	0.008

Table 4: Quantitative evaluation scores for each of the models on the different datasets. We use the average BLEU score and RougeL for comparison with previous baselines.

achieves higher performance than the GPT models in all cases. On FairytaleQA, we found that incorporating attribute into the model’s input did not significantly impact the RougeL or BLEU scores in comparison to the original variation. The T5 model achieves a higher RougeL score than the BART-QG model (0.527) fine-tuned in (Yao et al.,

Attribute	T5	T5-att	BART	BART-att	GPT	GPT-att
Action	0.562	<b>0.567</b>	0.371	0.335	0.291	0.269
Causal Relationship	<b>0.525</b>	0.408	0.364	0.400	0.257	0.307
Character	0.481	<b>0.516</b>	0.331	0.340	0.226	0.283
Feeling	<b>0.529</b>	0.507	0.424	0.403	0.347	0.354
Outcome Resolution	<b>0.478</b>	0.420	0.357	0.414	0.271	0.298
Prediction	<b>0.457</b>	0.410	0.324	0.303	0.119	0.203
Setting	<b>0.676</b>	0.667	0.486	0.438	0.428	0.346

Table 5: Breakdown of the RougeL scores by question attribute for the models fine-tuned on the FairytaleQA corpus.

FairytaleQA			
GT question	attribute	answer	Generated question
What happened after Finn lopped off the roots and branches of a fir-tree?	outcome resolution	He made a walking-stick of it and set out on his way to Oonagh.	T5:What did Finn do after lopping off the roots and branches? T5-attr:How did Finn get a fir-tree back? BART:What happened after Finn pulled up a fir-tree? BART-attr:What happened after Finn pulled up a fir-tree? GPT:What did Oonagh do while Finn lived at Tallinn? GPT-attr:What happened after Finn pulled up a fir-tree?
Where did Finn live?	setting	on Knockmany Hill	T5:Where did Finn live? T5-attr:Where did Finn live? BART:Where did Finn and his gigantic relatives live? BART-attr:Where did Finn live at this time? GPT:Where did Finn's large relatives live? GPT-attr:Where did Finn's large relatives live?
What will a farmer tell you about why the buckwheat looks black and singed?	action	The lightning did that	T5: Why did the farmer say the buckwheat looked black and singed? T5-attr: What made the buckwheat look black and singed? BART:Fire? BART-attr: Did it? GPT: What did the farmer write? GPT-attr: How did the farmer know about the black bear and its singed appearance?
TQA			
Air masses over the United States usually move toward the	-	northeast	T5:Air masses tend to move toward the BART:Where are air masses located? GPT:What direction does the air masses move?
Our own species evolved during the	-	Cenozoic Era	T5:mammals evolved during the BART:the age of mammals was GPT:Where did the men of the Cen section dwell?
Winds always blow from an area of higher to lower	-	pressure	T5:Air always flows from an area of higher pressure to an area of lower BART:Winds are generated by GPT:A higher pressure would cause a weaker wind to blow., a good question would be Why does the upper part of the North America sink? Hamburger

Table 6: Examples of questions generated by our models. The top three rows present examples of questions generated from the FairytaleQA dataset, while the bottom three rows depict examples of questions generated from the TQA dataset. We noted consistency in the performance of the T5 model across both datasets.

2022) on the test split. However, our fine-tuned BART model performs significantly worse than the one from (Yao et al., 2022).

## 5.1 Results on the FairytaleQA Corpus

As a whole, the T5 models produce more sensible and relevant questions than the other model variations on the FairytaleQA Corpus. When we take a look at some of the individual questions produced by the T5 model, we find that in some cases they are identical to the gold question or within one or two words. However, the automated metrics do not capture some critical semantic errors in the generated questions. In some cases, the T5 model hallucinates additional information in the questions. For example, for the anticipated question *Where did Granua live?*, both of T5 and T5-attr generate *Where did Oonagh and Granua live?*. Additionally,

the models sometimes switch the proper nouns between the subject and agent positions, changing the meaning of the gold question such as *What did Granua want from Oonagh?* to *What did Oonagh ask for from her sister?*. For these cases, we anticipate encoding more detailed discourse representations in the input, such as the use of named entity recognizers or abstract meaning representations could be highly beneficial.

### 5.1.1 By-Attribute Comparisons

Table 5 shows the by-attribute breakdown of RougeL scores for each of the model architectures. Similarly to the overall scores, the T5 variants outperform both BART and GPT, and BART variants outperform the GPT ones across all question attributes. All model variations have the highest scores for the *Setting* attribute questions. The generated samples for gold label questions such as

*Where did Finn live?* can be seen in Table 6. All of the generated questions start with ‘where’ or ‘when’, include the correct character, Finn, and the correct verb: ‘live’. The T5 model also achieves high scores on the *Action*, *Causal Relationship*, and *Feeling* questions. However, the BART baseline scores well on the attributes of *Outcome Resolution*, *Feeling*, and *Causal Relationship*, relative to its performance on the *Action* attribute. The BART model that encodes the attribute as part of the input outperforms the standard BART model for the *Outcome Resolution* and *Character* questions, but not for the other ones. The GPT model with attribute also achieves higher performance than the one without attribute for *Outcome Resolution* and *Character* questions suggesting that generating these questions may be more influenced by the type of question. The T5 model with attribute also outperforms the baseline variation for *Character* questions and *Action* as well. Unlike the BART and T5 variants, the GPT model with attribute exceeds the RougeL score of the majority of the questions. This suggests that GPT style models may benefit the most from including attribute information in the input step. One thing to consider when evaluating the attribute models is the fact that all of these models original pre-training procedures rely on input that does not include the attribute, so we are limited to exposing the model to this type of input in the fine tuning stage. We could hope to see performance improvements with attribute models with more attribute encoded data available for the fine-tuning stage.

## 5.2 Error Analysis of the TQA Dataset

As with the FairytaleQA dataset, we found that the T5 model outperformed both the BART and GPT models in terms of automated metrics. When we analyzed the generated questions, we observed that the T5 model incorporates more context into the questions than the other two models. Specifically, on this dataset, BART tended to produce shorter output questions or, in some cases, no output at all. In contrast, the GPT models frequently included unnecessary additions, such as one that randomly had the word ‘Hamburger’ appended to it. Refer to the last example of Table 6. The context passages included in this dataset require more specific concepts to be referenced, since generalizations may not be able to be made across passages in the text. For example, if a book is talking about how animals

in the great plains adapt to their environment, this information is not going to transfer to a passage about how animals in the tundra survive. Although these are both adaptations, we need the context specific values. This indicates the need to consider more complex models or additional ways of representing passage context. The use of a knowledge graph to represent facts introduced in the textbook could have significant benefit in this domain.

## 5.3 Cross-Corpus Comparison

All of the models tested performed significantly better on the FairytaleQA dataset than they did for the Textbook Question Answering dataset. There are a number of factors that could have contributed to this gap in performance. The Textbook Question Answering corpus was originally designed to help improve the visual question answering task, specifically for multiple choice questions. We have modified the dataset using automated methods to fit the open question generation task instead. Our preprocessing methods are automated and could use a human review to ensure that we are not trying to generate questions that require knowledge of other answers from the multiple choice setting. Furthermore, the corpus is a third of the size of the FairytaleQA Corpus. Both domains suffered from factual correctness errors with the model replacing key nouns or names in the generated question with incorrect ones. This is something that could potentially be addressed with the use of discourse relations that are embedding in input.

## 6 Limitations and Future Work

Future work on automated question generation for learning contexts could benefit from a number of potential research paths. In this paper, we tested three different architectures - but there are many more to be considered including those that incorporate knowledge graphs which have been shown to improve the richness and semantic correctness of generated questions (Bi et al., 2020). There is also room to explore different prompt strategies including a fill-in-the-blank approach which may be more appropriate for the TQA data. For the attribute models, we used the single task objective of question generation, but it would be worthwhile to explore jointly generating the question attribute and the question itself. Additionally, document level Abstract Meaning Representations with resolved coreferences has been shown to improve

the quality of knowledge based question generation (Kapanipathi et al., 2021). We also recognize that we focused on different context for the input, but not on the wide variety of generation strategies available for this task. On top of the variety of model architectures, we would like to evaluate a greater set of corpora that include additional topics such as history and economics. Reading comprehension is critical to these fields as well and there is limited, if any, research on question generation for these topics.

Additionally, in future work we will conduct evaluation with expert annotators to incorporate into more complex models. Ideally, we will have educators and students assess the output of our models for factual correctness, relevance, and fluency of the questions generated. This output can then be used to train an instruction fine-tuned model. In order to make a solution that is viable for the classroom, it is critical to think beyond the automated metrics and get real teacher feedback. This preliminary research demonstrates the potential for expanding automated question generation to multiple classroom subjects and the value of incorporating discourse information into different model architectures to produce high quality questions.

## 7 Conclusion

In this paper, we conduct an initial comparison of automated question generation architectures for narrative stories (fairytales) and science textbooks. For each corpus, we trained BART, GPT-2, and T5 models to see which would perform best in which context. Our results indicate that the T5 models achieve the highest scores in terms of automated metrics for both domains. The highest performing T5 model also outperforms the BART baseline for question generation on the FairytaleQA dataset put forth in (Xu et al., 2022b). We also evaluated the effectiveness of encoding question attribute information in different model architectures. We saw improvements in performance for both *Character* and *Outcome Resolution* questions when the attribute was included for multiple architectures suggesting that this information is beneficial for generating certain types of questions, but not all. Additionally, the inclusion of attribute information led to a more significant improvement across question types for the GPT architecture.

## 8 Acknowledgements

This research was supported by the NSF National AI Institute for Student-AI Teaming (iSAT) under grant DRL 2019805. The opinions expressed are those of the authors and do not represent views of the NSF. The authors would like to thank Dr. Alexis Palmer, Dr. Peter Foltz, Dr. James Martin, and Dr. Clayton Lewis for their insightful comments and suggestions throughout the experiments and paper writing process.

## References

- Sheng Bi, Xiya Cheng, Yuan-Fang Li, Yongzhen Wang, and Guilin Qi. 2020. Knowledge-enriched, type-constrained and grammar-guided question generation over knowledge bases. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2776–2786, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- 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](#).
- Khushnuma Grover, Katinder Kaur, Kartikey Tiwari, Rupali, and Parteek Kumar. 2021. Deep learning based question generation using T5 transformer. In *Advanced Computing*, pages 243–255. Springer Singapore.
- Pavan Kapanipathi, Ibrahim Abdelaziz, Srinivas Ravishankar, Salim Roukos, Alexander Gray, Ramón Fernández Astudillo, Maria Chang, Cristina Cornelio, Saswati Dana, Achille Fokoue, Dinesh Garg, Alfio Gliozzo, Sairam Gurajada, Hima Karanam, Naweed Khan, Dinesh Khandelwal, Young-Suk Lee, Yunyao Li, Francois Luus, Ndivhuwo Makondo, Nandana Mihindukulasooriya, Tahira Naseem, Sumit Neelam, Lucian Popa, Revanth Gangi Reddy, Ryan Riegel, Gaetano Rossiello, Udit Sharma, G P Shrivatsa Bhargav, and Mo Yu. 2021. [Leveraging Abstract Meaning Representation for knowledge base question answering](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3884–3894, Online. Association for Computational Linguistics.
- Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Are you smarter than a sixth grader? textbook question answering for multimodal machine

- comprehension. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.
- Tomáš Kočický, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.
- 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.
- Philippe Laban, Chien-Sheng Wu, Lidiya Murakhovska, Wenhao Liu, and Caiming Xiong. 2022. [Quiz design task: Helping teachers create quizzes with automated question generation](#). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 102–111, Seattle, United States. Association for Computational Linguistics.
- 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](#).
- 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.
- Chia-Wei Liu, Ryan Lowe, Iulian 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, Austin, Texas. Association for Computational Linguistics.
- 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](#). *CoRR*, abs/1611.09268.
- 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, 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](#).
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Manav Rathod, Tony Tu, and Katherine Stasaski. 2022. [Educational multi-question generation for reading comprehension](#). In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 216–223, Seattle, Washington. Association for Computational Linguistics.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. [Llama: Open and efficient foundation language models](#).
- Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordani, Philip Bachman, and Kaheer Suleman. 2017. [NewsQA: A machine comprehension dataset](#). In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada. Association for Computational Linguistics.
- Zhen Wang, Siwei Rao, Jie Zhang, Zhen Qin, Guangjian Tian, and Jun Wang. 2020. Diversify question generation with continuous content selectors and question type modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2134–2143, Online. Association for Computational Linguistics.
- Zichao Wang, Andrew S Lan, Weili Nie, Andrew E Waters, Phillip J Grimaldi, and Richard G Baraniuk. 2018. QG-net: a data-driven question generation model for educational content. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*, number Article 7 in L@S ’18, pages 1–10, New York, NY, USA. Association for Computing Machinery.
- Ying Xu, Dakuo Wang, Mo Yu, Daniel Ritchie, Bingsheng Yao, Tongshuang Wu, Zheng Zhang, Toby Li, Nora Bradford, Branda Sun, Tran Hoang, Yisi Sang, Yufang Hou, Xiaojuan Ma, Diyi Yang, Nanyun Peng, Zhou Yu, and Mark Warschauer. 2022a. [Fantastic questions and where to find them: FairytaleQA – an authentic dataset for narrative comprehension](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 447–460, Dublin, Ireland. Association for Computational Linguistics.
- Ying Xu, Dakuo Wang, Mo Yu, Daniel Ritchie, Bingsheng Yao, Tongshuang Wu, Zheng Zhang, Toby Li, Nora Bradford, Branda Sun, Tran Hoang, Yisi Sang, Yufang Hou, Xiaojuan Ma, Diyi Yang, Nanyun Peng, Zhou Yu, and Mark Warschauer. 2022b. [Fantastic questions and where to find them: FairytaleQA – an authentic dataset for narrative comprehension](#). In

*Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 447–460, Dublin, Ireland. Association for Computational Linguistics.

Bingsheng Yao, Dakuo Wang, Tongshuang Wu, Zheng Zhang, Toby Li, Mo Yu, and Ying Xu. 2022. It is AI’s turn to ask humans a question: Question-Answer pair generation for children’s story books. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 731–744, Dublin, Ireland. Association for Computational Linguistics.

Zhenjie Zhao, Yufang Hou, Dakuo Wang, Mo Yu, Chengzhong Liu, and Xiaojuan Ma. 2022. Educational question generation of children storybooks via question type distribution learning and event-centric summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5073–5085, Dublin, Ireland. Association for Computational Linguistics.

Wenjie Zhou, Minghua Zhang, and Yunfang Wu. 2019. Question-type driven question generation. 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 6032–6037, Hong Kong, China. Association for Computational Linguistics.

Bowei Zou, Pengfei Li, Liangming Pan, and Ai Ti Aw. 2022. Automatic true/false question generation for educational purpose. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 61–70, Seattle, Washington. Association for Computational Linguistics.

# A dynamic model of lexical experience for tracking of oral reading fluency

Beata Beigman Klebanov, Michael Suhan, Zuowei Wang, Tenaha O'Reilly

Educational Testing Service, Princeton NJ, USA

{bbeigmanklebanov, msuhan, zwang, toreilly}@ets.org

## Abstract

We present research aimed at solving a problem in assessment of oral reading fluency using children's oral reading data from our online book reading app. It is known that properties of the passage being read aloud impact fluency estimates; therefore, passage-based measures are used to remove passage-related variance when estimating growth in oral reading fluency. However, passage-based measures reported in the literature tend to treat passages as independent events, without explicitly modeling *accumulation* of lexical experience as one reads through a book. We propose such a model and show that it helps explain additional variance in the measurements of children's fluency as they read through a book, improving over a strong baseline. These results have implications for measuring growth in oral reading fluency.

## 1 Introduction

Teaching young children the skill of reading is one of the major tasks of an education system. In the U.S., a common solution to monitoring the development of reading skill is the periodic administration of oral reading fluency (ORF) tests, as fluency scores can serve as indicators of early literacy skills (Biancarosa et al., 2021; Hasbrouck and Tindal, 2017; Bernstein et al., 2017; Kim and Wagner, 2015; Pikulski and Chard, 2005). For example, the popular DIBELS test is administered three times a year. A specific passage is given in a particular grade at a particular time; e.g., the passage titled *Trees* is administered in the spring of 3rd grade (Biancarosa et al., 2021, p.106). ORF is typically measured as words read correctly per minute of oral reading (wcpm), which accounts for both accuracy and speed (Fuchs et al., 2001). Each passage is normed so that a student's performance can be mapped to a percentile score relative to peers.

One of the weaknesses of this system of monitoring is the need to administer specific, pre-set

assessment passages. First, time is taken away from reading for learning and pleasure to read for a test. Second, in striving for socio-culturally responsive assessment, one would want to give agency to teachers and students in choosing what to read, as choice and interest could enhance engagement and performance. Our reading app, RELAY READER,<sup>1</sup> addresses this weakness by letting students read different books aloud as a learning-and-pleasure activity and measuring ORF in the background. This solution also allows for continuous monitoring, which means that students receive frequent opportunities to demonstrate their skill.

A fundamental challenge in this endeavor is that since students read from a variety of stories throughout the year, it is not feasible to collect sufficient readings of every passage of every story to create norms. One might wonder why passage-specific norms are needed to begin with – won't students whose reading rate is 90 words per minute read any text at this rate? Alas, readers exhibit a distribution of wcpm across passages (Beigman Klebanov et al., 2020; Barth et al., 2014; Ardoin et al., 2005), as the reader's fluency is not the only factor accounting for some of the variance in the wcpm measurements.

In particular, passage effects are a known source of variance. A variety of measures proposed in the literature control for passage effects, including aspects of text complexity, genre, local discourse structure, and prosody (Beigman Klebanov et al., 2020; Barth et al., 2014). All these measures assume passages are independent – as they typically are in a testing context. However, passages in a book of fiction are not independent; there is continuity of characters and settings in a well-crafted narrative that create an immersive reading experience. This continuity could impact oral reading – while a reader might stumble on *Hogwarts* for the first time due to the word's unfamiliarity, the 50th

<sup>1</sup><https://relayreader.org/>

encounter is likely to be less challenging.

It is not only the rarest words that would become less challenging when mentioned many times; repeated encounters in general are known to produce faster readings (Bell et al., 2009). Had it been the case that the first chapter introduced all the word types to be used in the book and subsequent chapters repeated those in various combinations, one would expect a steady increase in the reading pace as the reader moves through the story. Such an increase would be only partially related to the improvement in the general ORF skill of the reader, since the increase relies heavily on repetition of the same limited vocabulary and will likely disappear, at least partially,<sup>2</sup> when unrelated text is read.

To the best of our knowledge, little is known about the relationship between repetition and story location. We hereby pose to the community a novel challenge of modeling the dynamic of a reader's lexical experience. We offer an initial exploration and show empirical results that suggest practical usefulness of further research in this area.

## 2 Surprisal

The reader does not start reading *Harry Potter* with a blank lexical slate, so-to-speak; the within-the-book experience is a continuation of an ongoing lexical experience that accumulates across prior reading materials (and other language experiences, with more or less direct connection to reading). We therefore model a reader's prior knowledge using a large corpus, with the book experience viewed as an addition to the corpus – dynamically, one word at a time.<sup>3</sup> For every word token in the book, we use a measure of surprisal at seeing this word at this location in the book – namely, surprisal given the starting background knowledge and the within-book experience up until the current location.

In prior research, surprisal is typically defined as log of inverse of probability (Tribus, 1961), that is, for a random variable  $Y$ , the surprisal of the value  $Y = y$  is given by  $\log_2 \frac{1}{P(y)}$ . In our case, the estimate of the probability  $P(y)$  for a word  $y$  is updated continuously as the student progresses through the book, token by token. Thus, words that are rare in general but frequent in the book will become less surprising as the reader moves

<sup>2</sup>It is possible that some of those heavily repeated words will also occur in another story.

<sup>3</sup>If the background corpus has 5,155,569 tokens, the first token in the book will be token number 5,155,570.

through the book, as their estimated probability will increase. Surprisal will be highest for completely new words appearing near the end of the book – this is the first occurrence in all the experience so far (background + book). In contrast, words that are generally more frequent than in the current book would become gradually more surprising, but the increase will be small, since a frequent word has accumulated a lot of prior occurrences and the impact of any new ones is relatively small. Thus, if a book generally has a lower frequency of the word *the* than the background corpus, *the* will become more surprising as one adds the book to their lexical experience, but since even a long single book is orders of magnitude shorter than a large corpus that models the background knowledge, the book will only have a small impact on the surprisal values of generally frequent words.

## 3 Experiment 1: Surprisal with respect to book location

### 3.1 Data sources

For the current study, we use two novels – *Harry Potter and the Sorcerer's Stone* by J. K. Rowling (**HP**) and *The Adventures of Pinocchio* by C. Colodi translated from Italian by Carol Della Chiesa (**Pinocchio**) – and four background corpora, in order to observe consistency (or not) of the patterns in the two books and robustness to variation in background corpora. The background corpora are:

**SFI** This corpus was compiled to allow estimation of word frequencies a student might have encountered after 12 years of schooling. The corpus covers a variety of text types, including samples from high school and college text books, classical and popular literature, non-fiction, biographies, speeches, periodicals, and encyclopedias (Breland et al., 1994).

**TASA3** The TASA corpus is a subset of SFI focused primarily on textbooks and other materials used in the US schools sampled by readability across grade levels (Zeno et al., 1995). Versions of this corpus have been used extensively to induce educationally relevant semantic spaces, e.g., Landauer et al. (1998). We use the cutoff for up to grade 3 readability,<sup>4</sup> in view of the study with 4th and 5th graders (Section 4).

<sup>4</sup>[http://wordvec.colorado.edu/word\\_embeddings.html](http://wordvec.colorado.edu/word_embeddings.html)



**BNC** The British National Corpus ([BNC Consortium, 2001](#)) has samples of written and spoken British English from a wide range of sources from the later part of the 20th century.

**SUBT** This corpus is comprised of subtitles from U.S. films from 1900–2007 and U.S. television series ([Brybaert and New, 2009](#)).

We use pre-existing unigram counts for each of the corpora, either as raw counts (for BNC, SUBT, TASA3) or deriving the probability estimates from the standard frequency indices (SFI) using the reversed estimated-to-standard frequency transformation<sup>5</sup> and the published total corpus sizes to induce estimated counts. Table 1 shows information about the various corpora.

Corpus	# tokens	# types (unique tokens)
TASA3	2,692,335	32,732
SFI	14,418,651	94,563
BNC	100,136,361	537,729
SUBT	49,719,560	73,609

Table 1: Corpora used to model prior lexical experience.

### 3.2 Data pre-processing

All background corpora were pre-processed to normalize British/American spelling and handle contractions and hyphenation. The tokenization process used for generating the unigram counts differed somewhat across corpora and we generally followed the tokenization practice of the given corpus when tokenizing the book as a continuation of experience following that corpus. For example, *can't* corresponds to two tokens *can n't* in BNC, whereas SFI only retains *can* as a token.

The next step is turning a book into a series of passages. Each book is split into consecutive passages of approximately 250 words (about one page): We add paragraphs to a passage as long as the total word count is under 250 words. Whether to add the next paragraph into the passage depends on whether there is a larger absolute difference

<sup>5</sup>We use the formula  $SFI = 10 (\log_{10} U + 4)$ , where SFI is the standardized frequency index and U is the estimated frequency per millions words using dispersion  $D = 1$ , following the definition in [Terzopoulos et al. \(2017\)](#), which differs slightly from that offered in [Breland et al. \(1994\)](#). SFI is the name of the standardized index and also of one of our corpora, since the paper that introduced the corpus was also the one to introduce the index ([Breland et al., 1994](#)).

from 250 with or without adding it. Thus, passages always contain full paragraphs. Passages do not cross chapter boundaries; if the last passage of a chapter is very short – less than 50 words – we discard it. Four chapter-final passages were discarded for HP and three for Pinocchio. Table 2 shows the descriptive statistics of the book data.

Book	# chapters	# passages	passage length mean (std)
HP	17	315	246.83 (28.82)
Pinocchio	19	162	241.75 (44.04)

Table 2: Descriptive information for the book data.

### 3.3 Measures

To represent surprisal patterns in a given passage, we experiment with four measures. We use average, median, and standard deviation (stdev) of token-level<sup>6</sup> surprisal estimates per passage and a high-percentile (97%) cut-off that captures the extent of surprisal of a few of the most surprising words in the passage. We expect the 97-percentile to capture invented or rare vocabulary – exactly the kind of words for which we expect the most impact upon multiple within-book encounters. Table 3 shows words above the 97% cut-off for three passages in the beginning, middle, and end of HP and Pinocchio, including surprisal estimates for each word using SFI as the background corpus.

### 3.4 Research questions

Our research questions are as follows. First, is it the case that the overall dynamic of surprisal within the book tends towards lower surprisal later in the book? Second, do we observe consistent patterns across (a) the two books, and (b) the different background corpora? If the patterns vary dramatically across corpora, this would underscore the need to model the target user’s prior reading profile in a more precise and personalized manner.

### 3.5 Results

Table 4 shows Pearson’s correlations between the surprisal measures and the serial number of the passage in the book. Our first research question is answered in the affirmative – it is the case that

<sup>6</sup>If a word occurs multiple times in a passage, each occurrence will get a slightly different surprisal value – a later mention would incorporate the experience of having seen the word earlier in the passage as well as of not having seen it since that prior mention; see examples in Table 3.

Loc	HP		Pinocchio	
	Word	Surp.	Word	Surp.
Early	dursley	21.20	geppetto	22.78
	dursley	20.97	polendina	22.78
	dudley	22.20	antonio	22.78
	dudley	21.78	geppetto	22.20
	dudley	21.46	geppetto	21.78
Middle	hermione	18.98	tremble	16.69
	hermione	18.93	dolphin	16.74
	overhearing	19.54	marionette	16.99
	gryffindor	19.20	dolphin	16.73
	seamus	20.33	gait	17.28
	filch	19.62	fro	17.41
	sneering	18.93	idle	16.86
Late	wardrobes	20.79	snail	17.0
	greener	19.54	lizard	16.06
	tidier	21.79	bravo	18.62
	bertie	21.79	mischief	16.65
	bott	21.79	deserve	15.96
	muggle	19.40	praise	15.91
	wizened	19.54	models	16.39
	muggles	19.54	obedience	17.64

Table 3: Words in early, middle, late HP and Pinocchio passages that are the top 3% surprisals for the passage. Words are listed in their book order: *Dursley* in row 2 occurs later in the passage than *Dursley* in row 1.

surprisal trends downwards as one moves through the book, for the two books and the four measures.

Measure	mean	median	stdev	97%
Corpus	HP/P	HP/P	HP/P	HP/P
TASA3	-.21/-.18	-.11/-.11	-.26/-.24	-.22/-.20
SFI	-.14/-.16	-.14/-.06	-.26/-.28	-.32/-.42
BNC	-.13/-.16	-.11/-.06	-.18/-.25	-.21/-.44
SUBT	-.16/-.14	-.16/-.08	-.11/-.23	-.29/-.28

Table 4: Pearson’s correlations between book location (serial number of the passage in the story) and surprisal measures. In each cell, the value for Harry Potter is shown first (HP), followed by Pinocchio (P).

To address the question of robustness towards variation in background corpora, Table 4 shows that the trends are generally similar across the four corpora. Figure 1 exemplifies the trends. The corpora are in agreement regarding the general trajectories even if the exact estimates of surprisal are different. Surprisal values are generally higher for the larger corpora, since the occurrence of new words is more surprising with more background experience. Interestingly, for HP, it is not the case that chapter 1 is consistently more surprising than the rest; chapters 5 (*Diagon Alley*) and 7 (*The Sorting Hat*) are more surprising. This makes sense with respect to the story – while some of the "normal" (*muggle*)

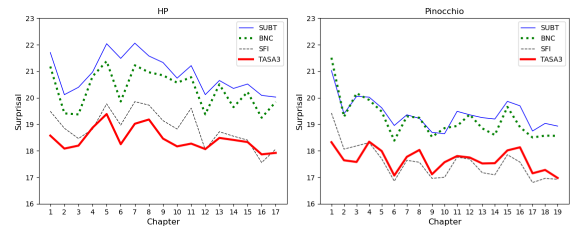


Figure 1: Average 97-percentile surprisal values per chapter across background corpora.

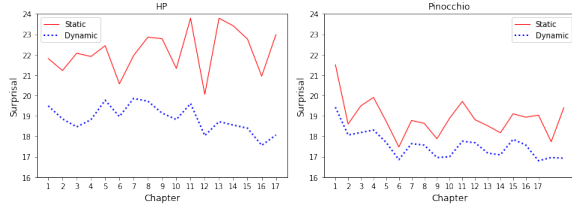


Figure 2: Average 97-percentile surprisals by chapter using static and dynamic SFI-based computations.

characters like the Dursley family are introduced in chapter 1, it is not until chapter 5 that the immersion in a very different, magic world happens, which is accompanied by a lot of rare or invented vocabulary related to magic artifacts (ch 5) and houses, teachers, and classes in a school of magic (ch 7). In contrast, the pattern for Pinocchio does show a drop after chapter 1, with more minor ups and downs later in the story.

To appreciate the difference between the measures discussed here and a ‘static’ surprisal calculation based on the background corpus only, without the dynamic recalculation following the token-by-token reading experience, Figure 2 plots the 97-percentile measure using the SFI background corpus. Without accounting for the within-book experience, some later chapters in HP have extremely high surprisal scores (chapters 13 and 14). The dynamic index shows, in contrast, that by that point in the story, life in a school of magic is somewhat business-as-usual, with these chapters being part of the general downwards trajectory. The discrepancy between the static and dynamic measures for the later HP chapters is such that the overall correlation with book location is actually *positive* for the static 97-percentile measure for all background corpora – in contrast to the universally negative correlations reported in Table 4 for the dynamic measures.

For the next experiment with 4th and 5th grade students, we used the TASA3 corpus to model background knowledge.

## 4 Experiment 2: Modeling fluency

### 4.1 Data

The oral reading data come from 35 students in grades 4 (12) and 5 (23) in an elementary school in New Jersey.<sup>7</sup> Students read on Amazon Fire 7 tablets with the RELAY READER app (previously called MY TURN TO READ, Madnani et al. (2019)) for up to 19 weeks, approximately three times a week for 20 minutes at a time, during the time generally set aside in the curriculum for independent reading. All the 35 students finished HP; those who finished earlier were provided the next book in the series in the paperback format. The students used consumer-level in-ear headphones with a built-in microphone.

When reading with the app, students took turns reading out loud consecutive passages of the book with a pre-recorded audiobook narrator. When splitting the text of a chapter into reading turns for the reader and the narrator, an algorithm as described in section 3.2 is used, with the target of 150 words per student turn and 200 words per narrator turn. The splitting is dynamic in that when the child first logs in on a given day or starts a new chapter, the narrator reads first starting from the current location, no matter who read last in the previous session, to ease the reader into the activity. Since students read at different rates, the daily starting locations varied and so did the passages read.

A set of 1,529 recordings with as many readers as possible per passage that span the beginning, middle, and end of each of the chapters were selected for the analysis, 67 passages in total with 100-170 words per passage (mean = 149.9, std = 17.5). Each reader contributed 13-64 readings (mean = 43.7, std = 13.2) and each passage was read by 15-33 children (mean = 22.8, std = 4.9). There were 60-111 recordings per chapter (mean = 90, std = 13). The recordings were transcribed by a professional agency. The transcribers were provided with the text of the passage and were asked to indicate any deletions, substitutions, and insertions as well as provide timestamps for the beginning and end of on-task speech. We then used the transcriptions to compute wcpm (the number of correctly read words divided by the time in minutes it took the child to read the passage).

<sup>7</sup>See the Ethics Statement for more detail.

### 4.2 Models

We now move to evaluating whether surprisal explains additional passage-based variance in wcpm, above and beyond baseline predictors. We fit linear mixed models using R's *lmer* function.

As a baseline, we use the model from Beigman Klebanov et al. (2020) where wcpm is modeled as a combination of passage and student random effects and a number of fixed effects: (1) the grade level of the student (to capture any systematic differences between grades); (2) a text complexity score produced by Text Evaluator (TE) (Napolitano et al., 2015); (3) a words-per-minute measurement of a "reading" generated by Apple's text-to-speech synthesizer (the Alex voice) to model variation in duration of different phonemes and reasonable inter- and intra-sentential pausing (TTS), and (4) the number of the chapter the passage is in. In the Beigman Klebanov et al. (2020) analysis, the coefficient of the chapter variable captures the average extent of improvement in oral reading fluency per chapter. Chapter is also used as a random slope to allow for different growth rates across participants. The model is specified using *lmer* syntax in equation 1; the coefficients are shown in the "Baseline" column of Table 5.

$$wcpm \sim (1|passage) + (chapter|student) + grade + TE + TTS + chapter \quad (1)$$

We next fit a model that is identical to the Baseline but has an additional fixed effect – the stdev of the surprisal values per passage, using the TASA3 corpus as background. The coefficients are shown in the "+Surprisal" column in Table 5. We show results with stdev index since models with 97% and mean did not converge and the model with median showed a similar pattern of results but worse fit than the model with stdev.

Table 5: Model estimates (with standard error). The values for TTS, TE, and Surprisal were standardized to  $\mu = 0$  and  $\sigma = 1$  and then entered into the model.

	Dependent variable: wcpm			
	Baseline		+Surprisal	
Grade 5	-0.83	(8.95)	-0.70	(8.95)
TTS	4.72***	(0.94)	3.37***	(0.89)
TE	-3.05**	(0.92)	-1.86*	(0.85)
Chapter	1.27***	(0.26)	1.09***	(0.25)
Surprisal			-3.39***	(0.76)
Constant	99.96***	(7.54)	101.41***	(7.51)

Note: \*p<0.05; \*\*p<0.01; \*\*\*p<0.001

We observe that surprisal is a significant predictor of wcpm, after controlling for complexity and prosody, with higher surprisal corresponding to slower reading. The Baseline model puts the amount of passage-based unexplained variance at 22.4; the number is reduced to 13 in the Baseline+Surprisal model, a reduction of 42%.

We also observe that the estimated rate of growth is somewhat reduced, from 1.27 additional wcpm per chapter to 1.09. This extent of growth is predicted after controlling for the within-book repetition of key book-specific vocabulary, so it might allow for a better estimate of the more generalizable part of the growth in fluency.

## 5 Related work

There exists a substantial body of work investigating the relationship between stand-alone properties of passages and the speed of reading. [Beigman Klebanov et al. \(2020\)](#) showed evidence that text complexity and prosody explain variance in children's wcpm. [Barth et al. \(2014\)](#) reviewed a variety of indices used to characterize the language in passages and found that text complexity, narrativity (the extent to which a passage is story-like rather than informational), and referential cohesion were predictive of wcpm, with complexity entering with a negative coefficient, while narrativity and cohesion enter the model with positive coefficients. Referential cohesion quantifies "the extent to which words overlap across sentences in the text" and is thus capturing an aspect of local, sentence-to-sentence predictability. A related but even more localized notion of predictability – within sentences rather than across sentences – was found to predict speedup in silent reading in adults; both syntactic and lexical immediate contexts were significant predictors ([Monsalve et al., 2012](#)). Given the findings, it is possible to manipulate the difficulty of a story by, for example, substituting shorter words instead of longer words or by repeating words across sentences.<sup>8</sup> These would, however, alter the language of the story and could reduce its literary quality and authenticity. In contrast, surprisal can be manipulated without changing the language by sequencing stories – having the first Harry Potter book in your prior reading experience would make a lot of the vocabulary in the second book less surprising.

---

<sup>8</sup>Indeed, text complexity is an explicit and quantitative design principle when creating texts for ORF assessments: "The Spache readability formula was used in creating and revising passages" ([Good and Kaminski, 2002](#), p.3).

Another related body of literature is the work on modeling word frequency distributions ([Piantadosi, 2014](#); [Baayen, 2001](#); [Katz, 1996](#)). In particular, the finding that various types of corpora, including single books, tend to exhibit certain consistent large-scale patterns of keyword burstiness is promising for generalization of findings such as ours across books ([Altmann et al., 2009](#); [Sarkar et al., 2005](#); [Montemurro and Zanette, 2002](#)).

The extensive work on language modeling in NLP, including the advances achieved with transformer models, can be brought to bear on modeling surprisal at various granularities (word, sentence, passage) and given various types of prior experience (model pre-training, fine-tuning). Furthermore, the assumption that an encounter results in reduction in surprisal for that word only is an over-simplification, as the literature on associative and semantic priming suggests that related words are also somewhat activated ([Pickering and Gambi, 2018](#); [Plaut and Booth, 2000](#); [Masson, 1995](#)). Transformer models were recently shown to exhibit certain priming effects themselves ([Lindborg and Rabovsky, 2021](#); [Misra et al., 2020](#)), making them a promising basis for modeling surprisal while accounting for priming effects. Our work with a word-level dynamic surprisal is just a first step.

## 6 Conclusion

In this paper, we presented a new NLP challenge coming out of the need to estimate the latent skill of oral reading fluency based on measurements of words read correctly per minute as readers move through a book using our electronic book reading app. Since the measurements are known to systematically depend on the properties of the passage, it is important to control for the passage-based variance in order to produce more precise skill estimates.

In particular, work presented here suggests that it is not only stand-alone properties of reading passages that are implicated in explaining slow-downs or speed-ups in oral reading, but also properties of a particular passage that have to do with its specific position in the reader's overall reading experience. As the reader reads through a book, they become more familiar with the special (invented or rare) vocabulary used in the book; this, in turn, could result in a speed-up in the reading. While the reader might be having an experience of increasing flu-

ency, some of the gain might be book-specific and therefore not generalize to the next book the developing reader tackles. Accurate tracking of oral reading fluency – a foundational reading skill that is a robust predictor of other skills such as comprehension – is a practical issue that will be helped by further research into dynamic models of a reader’s lexical experience.

## Limitations

The limitations of the findings in experiments 1 and 2 have to do with the relatively small scale of the study. We experimented with two books and, while the findings were broadly consistent, it could be that results would not generalize to other books. Experiment 2 was conducted with a specific group of readers in a specific context of implementation; studies with additional groups of readers are needed to evaluate generalization of the findings.

Another limitation of our experiments is that the dynamic model of lexical experience is evaluated only as an aggregate index per passage and not as a predictor for specific words or types of words. In particular, the model predicts a slight increase in surprisal of function words if their density in the story is generally lower than in the background corpus. This assumption may or may not be correct; further experimentation is necessary to evaluate the surprisal model in more detail. We thank a BEA reviewer for pointing out this limitation.

## Ethics Statement

RELAY READER, the reading app discussed in this paper, specifies Terms of Use and provides a link to Privacy Policy. In particular, the Terms of Use specify the legitimate uses of the data and commits to keeping the data of users-in-the-wild anonymous.<sup>9</sup>

For the book data, we used a public domain text of *The Adventures of Pinocchio* from Project Gutenberg and the text of *Harry Potter and the Sorcerer’s Stone* provided to us by the copyright holder<sup>10</sup> as a part of a license to use the book and the audiobook narration by Jim Dale in the app for a specified limited number of students; the students whose data is analyzed in Experiment 2 are within that cap.

<sup>9</sup><https://relayreader.org/terms>

<sup>10</sup>We did not alter anything in the HP book. For Pinocchio, we re-chaptered the original 36 short chapters of the story that we downloaded from Project Gutenberg into 19 longer chapters in order to better adjust to the turn-taking setup of RELAY READER.

The corpora used in the study are either broadly available for research purposes (BNC, SUBT) or have a more limited research and/or operational availability through contracts (TASA3, SFI).

The study during which oral reading data was collected from grade 4 and 5 students in a school in New Jersey was approved by the Institutional Review Board at our organization. Parental consent was obtained for students’ participation in the activity and for use of students’ data (including recordings, log data of the reading activity, and demographic information provided by the parents such as the grade data used in this study) for research.

The goal of this research is to improve the quality of assessment of oral reading by identifying factors that could impact fluency measurements that are not entirely due to the students’ developing skill and build models that would allow compensating for the impact of such factors. Accurate assessment of oral reading fluency controlling for text effects will benefit teachers and students in that the assessment can be done on a variety of texts, including different passages for different students, instead of using a single pre-set normed passage as in the current practice. This would give both teachers and students more agency in selecting reading materials based on interest and preference and will thus help assessment to be more socio-culturally responsive while still providing the measurement signal necessary to monitor skill progression.

## References

- Eduardo Altmann, Janet Pierrehumbert, and Adilson Motter. 2009. Beyond word frequency: Bursts, lulls, and scaling in the temporal distributions of words. *PLOS one*, 4(11):e7678.
- Scott Ardoin, Shannon Suldo, Joseph Witt, Seth Aldrich, and Erin McDonald. 2005. Accuracy of readability estimates’ predictions of CBM performance. *School Psychology Quarterly*, 20(1):1–22.
- Harald Baayen. 2001. *Word frequency distributions*, volume 18. Springer Science & Business Media.
- Amy Barth, Tammy Tolar, Jack Fletcher, and David Francis. 2014. The effects of student and text characteristics on the oral reading fluency of middle-grade students. *Journal of Educational Psychology*, 106(1):162–180.
- Beata Beigman Klebanov, Anastassia Loukina, JR Lockwood, Van Rynald Licalalde, John Sabatini, Nitin Madnani, Binod Gyawali, Zuowei Wang, and Jennifer Lentini. 2020. Detecting learning in noisy data:

- The case of oral reading fluency. In *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*, pages 490–495.
- Alan Bell, Jason Brenier, Michelle Gregory, Cynthia Girand, and Dan Jurafsky. 2009. Predictability effects on durations of content and function words in conversational English. *Journal of Memory and Language*, 60(1):92–111.
- Jared Bernstein, Jian Cheng, Jennifer Balogh, and Elizabeth Rosenfeld. 2017. Studies of a Self-Administered Oral Reading Assessment. In *Proceedings of SLATE 2017*, pages 180–184, Stockholm. KTH Royal Institute of Technology.
- Gina Biancarosa, Patrick Kennedy, Sunhi Park, and Janet Otterstedt. 2021. 8th Edition of Dynamic Indicators of Basic Early Literacy Skills (DIBELS®): Administration and Scoring Guide. Technical report, University of Oregon.
- BNC Consortium. 2001. [The British National Corpus, version 2 \(BNC World\)](#).
- Hunter Breland, Robert Jones, and Laura Jenkins. 1994. The college board vocabulary study. *College Board Report; Educational Testing Service Research Report*, 94(26).
- Marc Brysbaert and Boris New. 2009. Moving beyond Kučera and Francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for American English. *Behavior Research Methods*, 41(4):977–990.
- Lynn Fuchs, Douglas Fuchs, Michelle Hosp, and Joseph Jenkins. 2001. Oral reading fluency as an indicator of reading competence: A theoretical, empirical, and historical analysis. *Scientific Studies of Reading*, 5(3):239–256.
- Roland Good and Ruth Kaminski. 2002. DIBELS Oral Reading Fluency Passages for First through Third Grades. Technical report, University of Oregon, Eugene, OR.
- Jan Hasbrouck and Gerald Tindal. 2017. An update to compiled ORF norms. Technical report, Behavioral Research and Teaching, University of Oregon.
- Slava Katz. 1996. Distribution of content words and phrases in text and language modelling. *Natural Language Engineering*, 2(1):15–59.
- Young-Suk Kim and Richard Wagner. 2015. Text (oral) reading fluency as a construct in reading development: An investigation of its mediating role for children from Grades 1 to 4. *Scientific Studies of Reading*, 19(3):224–242.
- Thomas Landauer, Peter Foltz, and Darrell Laham. 1998. An introduction to Latent Semantic Analysis. *Discourse Processes*, 25(2-3):259–284.
- Alma Lindborg and Milena Rabovsky. 2021. Meaning in brains and machines: Internal activation update in large-scale language model partially reflects the N400 brain potential. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 43.
- Nitin Madnani, Beata Beigman Klebanov, Anastassia Loukina, Binod Gyawali, Patrick Lange, John Sabatini, and Michael Flor. 2019. [My turn to read: An interleaved E-book reading tool for developing and struggling readers](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 141–146, Florence, Italy. Association for Computational Linguistics.
- Michael Masson. 1995. A distributed memory model of semantic priming. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 21(1):3.
- Kanishka Misra, Allyson Ettinger, and Julia Rayz. 2020. Exploring bert’s sensitivity to lexical cues using tests from semantic priming. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4625–4635.
- Irene Monsalve, Stefan Frank, and Gabriella Vigliocco. 2012. Lexical surprisal as a general predictor of reading time. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 398–408.
- Marcelo Montemurro and Damián Zanette. 2002. Entropic analysis of the role of words in literary texts. *Advances in complex systems*, 5(01):7–17.
- Diane Napolitano, Kathleen Sheehan, and Robert Munkowsky. 2015. Online readability and text complexity analysis with TextEvaluator. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 96–100, Denver, Colorado.
- Steven Piantadosi. 2014. Zipf’s word frequency law in natural language: A critical review and future directions. *Psychonomic Bulletin & Review*, 21(5):1112–1130.
- Martin Pickering and Chiara Gambi. 2018. Predicting while comprehending language: A theory and review. *Psychological Bulletin*, 144(10):1002.
- John Pikulski and David Chard. 2005. Fluency: Bridge between decoding and reading comprehension. *The Reading Teacher*, 58(6):510–519.
- David Plaut and James Booth. 2000. Individual and developmental differences in semantic priming: empirical and computational support for a single-mechanism account of lexical processing. *Psychological Review*, 107(4):786.
- Avik Sarkar, Paul Garthwaite, and Anne De Roeck. 2005. A Bayesian mixture model for term re-occurrence and burstiness. In *Proceedings of the 9th Conference on Computational Natural Language Learning*, pages 48–55.

Aris Terzopoulos, Lynne Duncan, Mark Wilson, Georgia Niolaki, and Jackie Masterson. 2017. HelexKids: A word frequency database for Greek and Cypriot primary school children. *Behavior Research Methods*, 49(1):83–96.

Myron Tribus. 1961. Information theory as the basis for thermostatics and thermodynamics. *Journal of Applied Mechanics*, 28(1):1–8.

Susan Zeno, Stephen Ivens, Robert Millard, and Raj Duvvuri. 1995. *The educator's word frequency guide*. Touchstone Applied Science Associates.

# Rating Short L2 Essays on the CEFR Scale with GPT-4

Kevin P. Yancey and Geoffrey T. LaFlair and Anthony R. Verardi and Jill Burstein  
Duolingo

{kyancey, geoff, anthony.verardi, jill}@duolingo.com

## Abstract

Essay scoring is a critical task used to evaluate second-language (L2) writing proficiency on high-stakes language assessments. While automated scoring approaches are mature and have been around for decades, human scoring is still considered the gold standard, despite its high costs and well-known issues such as human rater fatigue and bias. The recent introduction of large language models (LLMs) brings new opportunities for automated scoring. In this paper, we evaluate how well GPT-3.5 and GPT-4 can rate short essay responses written by L2 English learners on a high-stakes language assessment, computing inter-rater agreement with human ratings. Results show that when calibration examples are provided, GPT-4 can perform almost as well as modern Automatic Writing Evaluation (AWE) methods, but agreement with human ratings can vary depending on the test-taker's first language (L1).

## 1 Introduction

Automated writing evaluation (AWE) systems are commonly used to evaluate test-taker writing. AWE systems are deployed on large-scale, high-stakes writing assessments used for admissions to higher education institutions, and for lower-stakes US state writing assessments that provide information about K-12 students' academic writing performance. These systems typically use feature-engineering approaches that include rule-based and statistical natural language processing (NLP) methods. NLP is used to extract features from essay writing responses that are characteristic of writing quality. Features may include errors in grammar and spelling, discourse structure, discourse coherence, vocabulary usage, and sentence variety. Features may be rule-based or statistically derived. Statistical model methods, such as straightforward linear regression, are used to train (build) AWE scoring models for high-stakes scoring of writing assessments. Detailed descriptions of systems are avail-

able for major systems, including e-rater®, Intelligent Essay Assessor™, Intellimetric®, and PEG (Shermis and Burstein, 2013), and Cambium's automated essay scoring system (Lottridge, in press).

Recent advances in language modeling with neural transformer architectures (OpenAI, 2023; Brown et al., 2020) have the potential to revolutionize AWE. These large language models (LLMs) demonstrate an incredible potential to analyze and evaluate text which has implications for the future of AWE. In addition, GPT's intuitive, text-based interface lowers barriers for use, potentially increasing accessibility and adoption of these tools for AWE. The assumptions about how LLMs – specifically GPT-4 – can be used for AWE tasks, such as automated scoring and feedback need to be evaluated to determine how we can use them beneficially, and particularly to ensure that they can be used in a fair and ethical manner (Burstein, 2023).

Previous research evaluated GPT-3.5 for essay scoring tasks in an L2 context (Mizumoto and Eguchi, 2023). In this paper, we evaluate GPT-4 for a similar task, comparing it to GPT-3.5, human judgement, and a strong baseline using current AWE methods. We also explore various aspects that affect the accuracy of GPT's ratings, and its fairness across gender and L1.

## 2 Data

For our experiments, we used a human-rated dataset consisting of short essay responses collected as part of the Duolingo English Test, a high-stakes test of English for L2 learners. For this essay task, test-takers are given a short written prompt randomly selected from an item bank of about 700 items. Test-takers have 5 minutes to provide their essay response to the prompt. Two human raters used a scoring rubric aligned with the Common European Framework of Reference (CEFR) (Council of Europe, 2001).

We started by sampling 10,000 responses from



test sessions that took place over a 10-month period, controlling for L1 and gender. For L1, we limited responses to 7 of the most common L1 languages for the test, which also captures a broad range of language families: Arabic (ara), Mandarin Chinese (cmn), Telugu (tel), English (eng)<sup>1</sup>, Spanish (spa), Gujarati (guj), and Bengali (ben). To ensure all CEFR levels were well represented in the final dataset<sup>2</sup>, we used a simple CEFR classifier that uses logistic regression and NLP features to roughly estimate the CEFR level of each response. For the final dataset, we randomly sampled an equal number of responses for each combination of L1, gender, and estimated CEFR level from the 10,000 test sessions.

The scoring rubric was aligned to the CEFR scale and assessed each response based on its content, coherence, vocabulary, and grammar. The rubric instructed raters to assign each essay one of eight rating categories: six based on the CEFR scale, and two “unscorable” categories for minimal responses (e.g., provides no response or says they can’t answer the question) and bad-faith responses (e.g., off-topic or nonsensical). The full rubric is provided in Appendix B.

Based on this rubric, two assessment researchers developed a set of calibration examples by collectively rating 676 essays, 180 of which were rated by both. The rubric and calibration examples were provided to two new human raters, who collectively rated 1,961 new essays, including a random sub-sample of 389 essays that were rated by both. Both new human raters were trained by one of the original assessment researchers and inter-rater agreement was routinely checked. Raters were provided feedback to help with calibration when necessary. The final Quadratic Weighted Kappa (QWK) between the two raters was 0.87. Ratings were roughly normally distributed (see Figure 1), with  $\sim 53\%$  of essays receiving a rating of B1 or B2 and only  $\sim 12\%$  getting a rating of A1 or C2.

<sup>1</sup>Test-takers who identify their L1 as English may come from countries where English is an official language, such as India. These test-takers are required to take an English language proficiency test to attend an English-medium institution abroad.

<sup>2</sup>In particular, the DET test-taker population’s proficiencies follow a unimodal distribution around the B1/B2 CEFR levels (Cardwell et al., 2022), and so uniform random sampling would have resulted in too few A1 and C2 essay responses being included in the dataset.

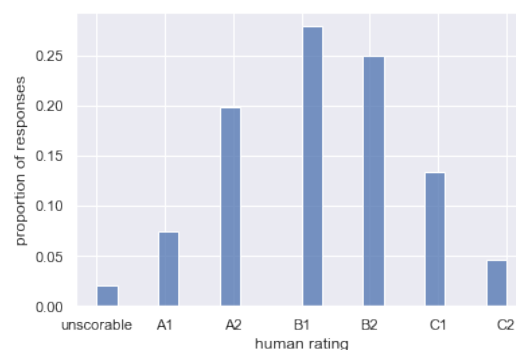


Figure 1: Distribution of Human Ratings by Raters 1 and 2

## 2.1 Methodology

In our experiments, we used the ChatGPT API to rate these short essay responses, comparing them to human judgements using the same rubrics.

In the system message, we instructed GPT to rate each provided essay in one of eight rating categories: one of the six CEFR levels or one of the two unscorable categories, [No-Response] and [Nonsense/Off-Topic]. In the default setting, we provided specific criteria the two unscorable categories, but not for CEFR levels<sup>3</sup>. See Appendix C for details.

In addition to the system message, we also provided GPT with varying numbers of calibration examples. These examples were randomly sampled from the set of 180 essays that were double-rated by assessment researchers where both researchers agreed on the same rating. The same number of examples were provided for each of the eight rating categories. We tested providing up to the maximum number of calibration examples that would fit into each model’s token limit (generally two per category for GPT-3.5 and four per category for GPT-4)<sup>4</sup>. To avoid any possible interaction between essays, we used a fresh GPT conversation to rate each essay.

<sup>3</sup>Querying GPT-4 easily shows that it already has some built-in knowledge of CEFR, presumably from its massive training corpora, and can even provide CEFR descriptors for various language skills verbatim, if prompted. So, it was reasonable to evaluate GPT’s ability to apply CEFR rating categories accurately without a rubric. The same is not true for the unscorable rating categories, and preliminary experiments showed that GPT applied the unscorable labels much too broadly if their criteria weren’t elaborated in the instructions to GPT.

<sup>4</sup>Note that this token limit applies to the entire GPT conversation, not just a single turn within the conversation, and thus this puts a hard limit on the number of calibration examples that can be provided.

Once all ratings were collected, we tabulated them on a scale of 0 – 6: assigning a 0 for both unscorable categories, and a score 1 – 6 for the CEFR levels. We then computed the inter-annotator agreement between GPT and rater 1 ( $n=1,175$ ), computing 90% confidence intervals using bootstrapping and comparing this to the agreement between the two human raters. We also compared our results to two baselines: a machine learning (ML) classifier using only the response’s character length, and a strong baseline representative of current AWE methods that use feature engineering and statistical modeling (Attali and Burstein, 2006; Foltz et al., 1999). The strong AWE baseline, which is used to score writing responses on the Duolingo English Test, uses XGBoost (Chen and Guestrin, 2016) and is trained on hundreds of thousands of short essay responses using 85 research-based linguistic features covering a wide range of writing sub-skills, including cohesion, grammatical complexity, lexical sophistication, grammatical and lexical accuracy, length, and relevance. A more detailed breakdown of these features are provided in Appendix A.

### 3 Experiments

We conducted three experiments. The first evaluates both GPT-3.5 and GPT-4 with a minimal rubric and up to the maximum number of calibration examples that fit within the GPT model’s token limit. The second experiment evaluates various prompt engineering strategies for improving performance. The third experiment explores GPT-4’s fairness properties across gender and L1.

#### 3.1 Experiment 1: Calibration Only

In this first experiment, we evaluated GPT’s ability to rate essay responses on the CEFR scale when provided only a minimal rubric (as described in Appendix C) and varying numbers of calibration examples.

Figure 2 shows the QWK between GPT and the first human rater, depending upon the model used and the number of calibration examples provided. When no calibration examples were provided, neither GPT-3.5 nor GPT-4 even outperform the baseline classifier using character length only. However, by providing just one calibration example for each rating category, GPT-4 almost matches the performance of the AWE baseline (QWK 0.81 vs 0.84,  $p < 0.1$ ). Providing additional examples did not result in significant improvement. GPT-3.5, on the

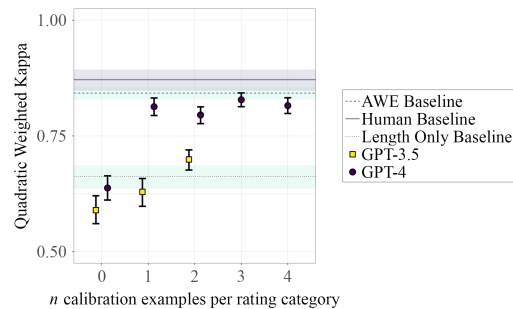


Figure 2: Human–GPT agreement when only calibration examples are provided (90% confidence intervals shown)

other hand, did not improve much when provided calibration examples, and only outperformed the length-only baseline when provided two calibration examples per rating category (i.e., the maximum possible with GPT-3.5’s limit of 4,096 tokens).

The confusion matrices in Figure 3 provide more insight. We see that when no examples were provided, both versions of GPT were generally able to identify unscorable responses, and did tend to assign slightly higher ratings to better essays, but mainly rated essays in the B1 – B2 range. When provided calibration examples, GPT-4 learned to use the full range of CEFR levels, but struggled to distinguish between adjacent CEFR levels compared to humans, especially for CEFR level B2. GPT-3.5, on the other hand, improves only slightly when provided calibration examples.

#### 3.2 Experiment 2: Prompt Engineering

In our second experiment, we tested two strategies for improving the performance of GPT-4:

**Detailed Rubric** - In the system message, we replaced the minimal rubric used in the previous experiment with a detailed rubric that described the criteria for each CEFR level (see Appendix C).

**Require Rationale** - In the system message, we asked GPT to provide a rationale before providing its rating in order to elicit a chain of reasoning, which has been shown to improve the the ability of LLMs to perform complex tasks (Wei et al., 2022). This also meant providing rationales for the calibration examples, which could help GPT-4 better understand the reason for each example’s rating.

Both of these techniques required significantly more token-space for the input prompt and thus lim-

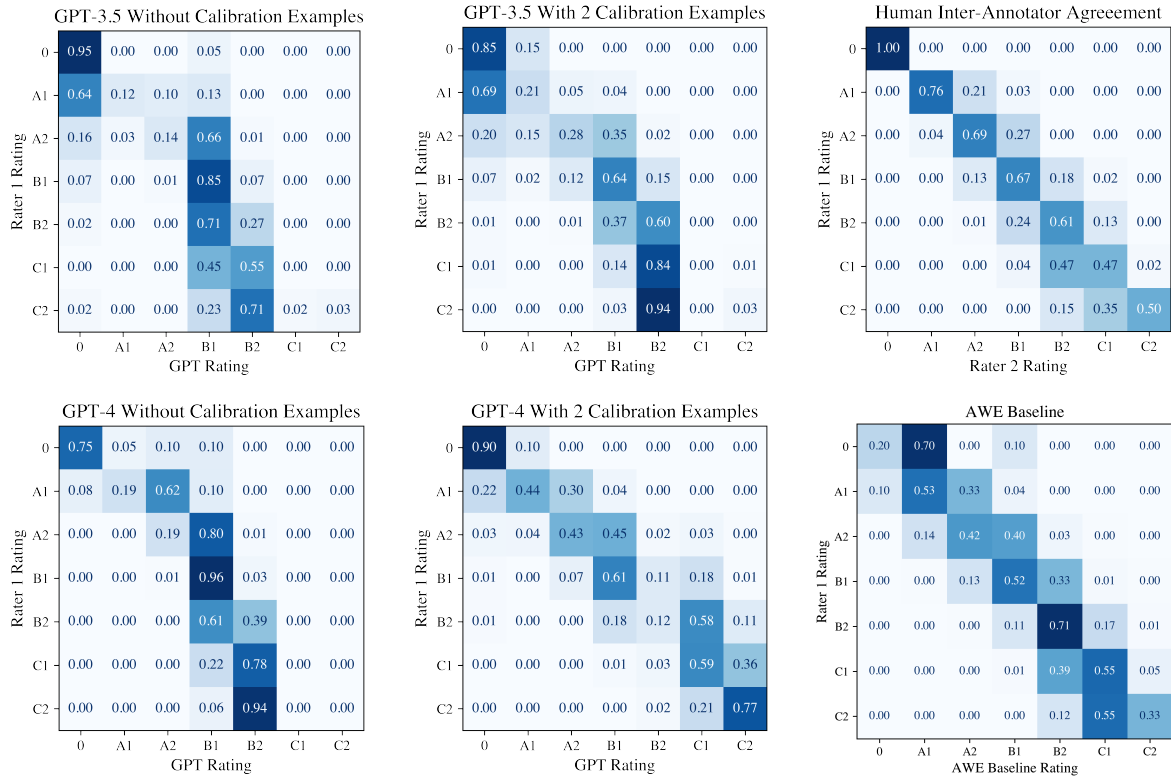


Figure 3: Confusion Matrices (Normalized by Rater 1's Rating)

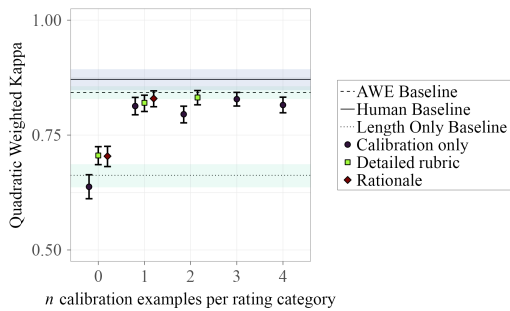


Figure 4: Human-GPT-4 agreement when various prompt engineering techniques are applied (90% confidence intervals shown)

ited the number of calibration examples that could be provided. Only up to two per rating category could be provided when using a detailed rubric, and only up to one per rating category when requiring rationales.

As seen in Figure 4, these strategies contributed substantial lift in performance when not providing calibration examples, but when at least one calibration example per rating category was provided, these techniques contributed negligible benefit.

### 3.3 Experiment 3: Fairness

Ensuring that raters do not show systematic bias that can affect scoring accuracy due to background characteristics of test-takers, such as gender or L1, is an important step in rater analysis with human raters (Jin and Eckes, 2022). This is also a needed step in developing AWE systems. To investigate the extent to which GPT-4's ratings are fair, we evaluated its performance for each gender and each of the L1 languages in the dataset.

To maximize statistical power and ensure that the analysis is not biased by a single human rater, we used all essays rated by any one of the raters or researchers in our dataset, except the 180 essays that were double-rated by the two researchers, which were reserved for calibration examples. The resulting dataset included 2,457 essays, roughly equally distributed among both genders and all L1s.

We found no significant differences in performance by gender, and while GPT-4's ratings were slightly positively biased compared to human ratings overall (by about +0.15 CEFR levels), this bias did not vary significantly by any gender or L1 ( $p > 0.10$ ).

However, we did find that GPT-4 had less agreement with human ratings for essays written by L1

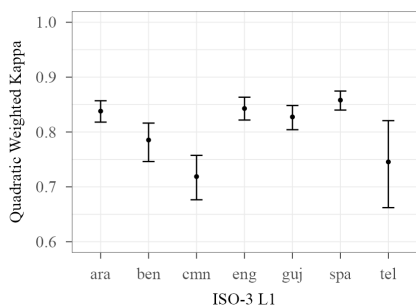


Figure 5: GPT-4 QWK by test-taker L1

speakers of some languages compared to others: QWK was lowest for L1 speakers of Telugu (tel) at 0.66 and highest for L1 speakers of Spanish (spa) at 0.89. A more detailed analysis showed that some of the differences in agreement by L1 was explained by differences in the distribution of human ratings for those L1s. The standard deviation of human ratings by L1 ranged from 1.04 for Telugu (tel) to 1.56 for Arabic (ara). Those L1s with narrower distributions of human ratings had a greater proportion of essays rated in categories for which GPT-4 had lower rates of agreement overall, such as B2, and thus brought down the QWK for those L1s.

We assume that the differences in the distribution of human ratings by L1 reflect systematic errors in the CEFR classifier used in sampling (see Section 2) and possibly differences in our underlying test-taker population. Thus we controlled for these distribution differences by recomputing QWK for each L1 using importance sampling so that all L1s would have the same effective distribution of human ratings. The results are shown in Figure 5. Even after the importance sampling correction is applied, GPT-4’s ratings agreed less with human ratings for responses written by L1 speakers of Mandarin Chinese (cmn), Telugu (tel), and Bengali (ben) compared to those written by L1 speakers of Spanish (spa). It is possible that essays of some L1s are harder to distinguish and thus have less reliable human ratings, but our dataset does not consist of a sufficient number of double-rated essays to investigate this hypothesis, so we leave this for a future work.

## 4 Conclusion

We showed that unlike GPT-3.5, GPT-4 is able to attain performance similar to conventional Automated Writing Evaluation (AWE) models when rating short L2 essays. GPT-4 only required one calibration example per rating category to achieve

near optimal performance, but other prompt engineering techniques we tried were not very helpful. Furthermore, when assessing fairness with respect to the test-taker’s gender or L1, we found that while GPT-4 did not show bias in favor of any one group, it showed significantly less agreement with human ratings for some L1s. It is unclear whether this is due to the reliability of GPT-4 or that of the human ratings themselves. More research is needed to understand this discrepancy and its implications for fairness. Future research may also explore other prompt engineering strategies for improving GPT-4’s performance at this task, or potentially fine-tuning GPT-3.5, enabling one to leverage dramatically more training data than what can be provided in a prompt. Perhaps most excitingly, future work may explore GPT-4’s potential for providing feedback aligned to essay scoring: a task for which GPT-4 seems particularly well suited.

## Acknowledgements

We thank the researchers and raters who contributed to building the dataset, and the reviewers who reviewed our paper and provided valuable feedback, particularly JR Lockwood, Ben Naismith, Klinton Blicknell, and Alina von Davier.

## References

- Yigal Attali. 2011. A differential word use measure for content analysis in automated essay scoring. *ETS Research Report Series*, 2011(2):i–19.
- Yigal Attali and Jill Burstein. 2006. Automated essay scoring with e-rater® v. 2. *The Journal of Technology, Learning and Assessment*, 4(3).
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- CJ Bryant, Mariano Felice, and Edward Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. Association for Computational Linguistics.
- Jill Burstein. 2023. [Responsible ai standards](#).
- Ramsey Cardwell, Geoffrey T LaFlair, and Burr Settles. 2022. Duolingo english test: Technical manual.
- Tianqi Chen and Carlos Guestrin. 2016. [XGBoost: A scalable tree boosting system](#). In *Proceedings of the 22nd ACM SIGKDD International Conference on*

*Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA. ACM.

Peter W Foltz, Darrell Laham, and Thomas K Landauer. 1999. The intelligent essay assessor: Applications to educational technology. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, 1(2):939–944.

Kuan-Yu Jin and Thomas Eckes. 2022. Detecting differential rater functioning in severity and centrality: The dual drf facets model. *Educational and Psychological Measurement*, 82(4):757–781.

S. Lottridge. in press. *Applications of transformer neural networks in processing examinee text*, MARCES Book Series. University of Maryland Press.

Philip M McCarthy and Scott Jarvis. 2010. Mtd, vocd-d, and hd-d: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior research methods*, 42(2):381–392.

Danielle S McNamara and Arthur C Graesser. 2012. Coh-metrix: An automated tool for theoretical and applied natural language processing. In *Applied natural language processing: Identification, investigation and resolution*, pages 188–205. IGI Global.

Atsushi Mizumoto and Masaki Eguchi. 2023. Exploring the potential of using an ai language model for automated essay scoring. *Research Methods in Applied Linguistics*, 2(2):100050.

Council of Europe. Council for Cultural Co-operation. Education Committee. Modern Languages Division. 2001. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment*. Common European Framework of Reference for Languages: Learning, Teaching, Assessment. Cambridge University Press.

OpenAI. 2023. *Gpt-4 technical report*.

Marek Rei and Ronan Cummins. 2016. Sentence similarity measures for fine-grained estimation of topical relevance in learner essays. *arXiv preprint arXiv:1606.03144*.

Sarah E Schwarm and Mari Ostendorf. 2005. Reading level assessment using support vector machines and statistical language models. In *Proceedings of the 43rd annual meeting of the Association for Computational Linguistics (ACL'05)*, pages 523–530.

Mark D Shermis and Jill Burstein. 2013. *Handbook of automated essay evaluation*. NY: Routledge.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Menglin Xia, Ekaterina Kochmar, and Ted Briscoe. 2019. Text readability assessment for second language learners. *arXiv preprint arXiv:1906.07580*.

## A AWE Baseline Model Features

Here we provide a more detailed breakdown of the features used in our AWE baseline:

- 13 cohesion features, including overlap features and coreference counts (McNamara and Graesser, 2012)
- 3 grammatical complexity features, including max/mean dependency tree depth and mean sentence length (Schwarm and Ostendorf, 2005)
- 7 lexical sophistication features measuring the proportion of words at each CEFR level (including an out-of-vocabulary category for words that could not be found in the CEFR dictionary) (Xia et al., 2019)
- 51 lexical and grammatical accuracy features, measuring the error rates across a wide variety of error types (Bryant et al., 2017)
- 4 features using n-gram models over word-forms, lemmas, part-of-speech, and dependency tags to measure differential use of vocabulary and grammar across test-takers of different proficiency levels (Attali, 2011)
- 3 length features, including number of characters, words, and sentences
- 2 lexical diversity features derived from the Measure of Textual Diversity (MTLD) (McCarthy and Jarvis, 2010)
- 1 vocabulary control feature using n-gram models to measure idiomatic use of vocabulary
- 1 relevance feature, computed using IDF weighted word embeddings between the prompt and the response (Rei and Cummins, 2016)

## B Scoring Rubric

Below are the criteria for each rating that were used in the rubric provided to human raters, and the system message prompts provided to ChatGPT (where applicable).

- C2** The response fully achieves the task requirements: (1) the response is clear, relevant, fully developed, and is written in an appropriate

style (2) the response is smoothly-flowing, coherent, and cohesive throughout; (3) vocabulary (including collocations and idiomatic language) is accurate, appropriate, and precise; and (4) a wide range of grammatical structures are flexibly used, and there are no grammatical errors other than slips characteristic of expert speakers. Does the response have an excellent effect on the reader, such that the writer communicates their position/describes the image extremely effectively and in detail, there is no strain on the reader, and a very high level of language is used consistently throughout?

- C1** The response achieves the task requirements: (1) the response is clear, relevant, appropriately developed, and is written in an appropriate style (2) the response is well-structured, coherent, and cohesive; (3) vocabulary (including collocations and idiomatic language) is accurate, appropriate, and demonstrates a broad range; and (4) a wide range of grammatical structures are used, and grammatical errors are rare. Does the response have a very good effect on the reader, such that the writer communicates their position/describes the image clearly and effectively at some length, with a high level of language used consistently throughout other than minor lapses which do not impact the communicative effect?
- B2** The response mostly achieves the task requirements: (1) the response is mostly clear, relevant, developed, and written in an appropriate style (2) the response is generally well-structured, coherent, and cohesive despite occasional lapses; (3) vocabulary (including collocations and idiomatic language) is generally accurate and appropriate to the task; and (4) a range of grammatical structures are used, and grammatical errors usually do not impact communication. Does the response have a good effect on the reader, such that the writer communicates their position/describes the image fairly clearly and with some detail, with a level of language that allows them to successfully complete the task despite inaccuracies?
- B1** The response partially achieves the task requirements: (1) the response is not always clear, relevant, developed, or written in an appropriate style (2) the response is somewhat organized

but may lack coherence or cohesion at times; (3) vocabulary (including collocations and idiomatic language) is generally clear but limited; and (4) a limited range of grammatical structures are used with some errors which may impact communication. Does the response have a satisfactory effect on the reader, such that the writer communicates their position/describes the image despite lapses, with a level of language that allows them to generally complete the task despite errors?

- A2** The response minimally achieves the task requirements and may be somewhat off-topic or underlength: (1) the response is limited to simple descriptions/personal opinions and topics and may be unclear, irrelevant, or written in an inappropriate style or format (2) the response uses some simple cohesive devices but may be repetitive or incoherent at times; (3) vocabulary is limited and often inaccurate or unclear; and (4) grammar structures are basic and there are frequent errors which may impact communication. Does the response have a poor effect on the reader, such that the writer communicates only basic impressions or opinions/a basic description, with a level of language that allows them to only minimally complete the task despite numerous errors?

- A1** The response does not achieve the task requirements and may be off-topic or very underlength: (1) the response is limited to simple personal information and does not present a position/describe the image. Ideas are often unclear or irrelevant. (2) the response does not demonstrate organizational features and is composed of isolated phrases and sentences; (3) vocabulary is very limited, inaccurate, and is insufficient for the task; and (4) only basic grammatical structures are produced and errors predominate. Does the response have a very poor effect on the reader, such that the writer does not communicate a relevant position/adequately describe the image, with a level of language that does not allow them to successfully complete the task?

**No-Response** There is no response, it is very minimal, or the test-taker indicates that they cannot answer the question (e.g., “I don’t understand”, “Sorry my English is bad”, etc.).

**Nonsense/Off-Topic** The test-taker does not respond to the prompt in good faith, repeats the prompt without responding to it, or intentionally goes off-task in an attempt to “trick” the system (e.g., by writing random words, writing in a non-English language, writing random strings of letters, or giving a memorized off-topic response).

## C GPT Prompts

The wording and design of the prompts provided to GPT can affect its performance. In this appendix, we provide the exact details of each prompt we used.

For our purposes, there are two components to the GPT prompts: the system message and the conversation turns. The system message tells ChatGPT the role it is playing in the conversation, and helps set its behavior during the interaction. For the system messages, we used two different messages, depending on whether the rubric was provided or not.

When providing a minimal rubric to GPT without asking for a rationale, we used the following message:

```
You are a rater for writing responses on
a high-stakes English language exam
for second language learners. You
will be provided with a prompt and
the test-taker's response.
```

```
Ratings are based on the CEFR scale.
Each rating should be one of the
following: [A1], [A2], [B1], [B2], [
C1], [C2], [Nonsense/Off-Topic], or
[No-Response].
```

```
You should assign a [No-Response] rating
if:
- There is no response to assess.
- There is no or very minimal response.
- The test-taker indicates they cannot
answer the question (e.g., I don't
understand, Sorry my English is bad,
etc.).
```

```
You should assign a [Nonsense/Off-Topic]
rating if:
- The test-taker is not responsive to
the prompt in good faith:
- The test-taker repeats the prompt but
does not respond to it.
- The test-taker intentionally goes off-
task in some way to 'trick' the
system, e.g., by writing random
words, writing in a non-English
language, writing random strings of
letters, or giving a memorized off-
topic response.
```

```
You should reply to each response with
just your rating: do not explain or
justify it.
```

When the rubric was provided to GPT, we used the message below, which adds the descriptions for each CEFR level. We used the same descriptions as defined in Appendix B, so we elide them here, replacing them with a comment between angled brackets <>, for brevity.

```
You are a rater for writing responses on
a high-stakes English language exam
for second language learners. You
will be provided with a prompt and
the test-taker's response.
```

```
Ratings are based on the CEFR scale.
Each rating should be one of the
following: [A1], [A2], [B1], [B2], [
C1], [C2], [Nonsense/Off-Topic], or
[No-Response].
```

Scoring Criteria:

```
For each CEFR rating, there is a
description which addresses relevant
aspects of language related Content,
Discourse, Vocabulary, and Grammar.
When assigning a score, the overall
holistic impression should be
considered it is not necessary for
a test-taker to achieve all of the
positive characteristics of a grade
as long as overall the descriptor is
the best match.
```

```
Rating: [C2]
Description: <See description in
Appendix A above>
```

```
<Repeated for ratings C1 - A1>
```

```
You should assign a [No-Response] rating
if:
- There is no response to assess.
- There is no or very minimal response.
- The test-taker indicates they cannot
answer the question (e.g., I don't
understand, Sorry my English is bad,
etc.).
```

```
You should assign a [Nonsense/Off-Topic]
rating if:
- The test-taker is not responsive to
the prompt in good faith:
- The test-taker repeats the prompt but
does not respond to it.
- The test-taker intentionally goes off-
task in some way to 'trick' the
system, e.g., by writing random
words, writing in a non-English
language, writing random strings of
letters, or giving a memorized off-
topic response.
```

```
You should reply to each response with
just your rating: do not explain or
justify it.
```

In both cases, we explicitly instructed GPT not to explain or justify its responses, to ensure that a definitive rating that could be parsed and used in the evaluation would be provided. When we experimented with requesting rationales as described in Experiment 2, we replaced the last line with the following:

```
You should reply to each response with
your rationale and rating in the
following format:
```

```
Rationale: <<<Your rationale here.>>>
```

```
Rating: [<<<Your rating here.>>>]
```

The conversation turns were used to provide GPT with the essay to be rated, and to elicit a rating. It was also used to provide GPT with calibration examples, when applicable. In both cases, we used the same format.

The user message provides the essay prompt and the test-taker's response. As recommended by OpenAI, both are surrounded in triple-quotes.

```
Prompt: """
<Essay prompt placed here.>
"""
```

```
Response: """
<Essay response placed here.>
"""
```

The assistant response message following each user message would simply contain the rating in square brackets (e.g., [B2] or [Nonsense/Off-Topic]). In most cases, GPT would prefix its response with `Rating:`, which we simply dropped.



# Towards automatically extracting morphosyntactical error patterns from L1-L2 parallel dependency treebanks

Arianna Masciolini and Elena Volodina and Dana Dannélls

Språkbanken Text

Department of Swedish, Multilingualism, Language Technology

University of Gothenburg

firstname.lastname@gu.se

## Abstract

L1-L2 parallel dependency treebanks are UD-annotated corpora of learner sentences paired with correction hypotheses. Automatic morphosyntactical annotation has the potential to remove the need for explicit manual error tagging and improve interoperability, but makes it more challenging to locate grammatical errors in the resulting datasets. We therefore propose a novel method for automatically extracting morphosyntactical error patterns and perform a preliminary bilingual evaluation of its first implementation through a similar example retrieval task. The resulting pipeline is also available as a prototype CALL application.

## 1 Introduction

L1-L2 parallel dependency treebanks are corpora where sentences produced by learners of a second language (L2), paired with native-like (L1) correction hypotheses, are annotated following the Universal Dependencies (UD) standard (Nivre et al., 2020). This data format, proposed by Lee et al. (2017), has interoperability as its main goal: UD provides a uniform annotation layer across different languages and its fine-grained morphosyntactical analysis is meant to make explicit error tagging unnecessary, preventing the incompatibilities that arise from the use of project-specific taxonomies. In addition, the availability of increasingly reliable dependency parsers can significantly speed up, if not completely automate, the annotation process.

Putting L1-L2 treebanks into use, however, requires effective ways to extract information from them. Errors, explicitly marked in most learner corpora, are for instance not straightforward to identify in such datasets. In this paper, we report on ongoing work on this problem, focusing on morphosyntax. In particular, we propose a novel approach to locate error-correction pairs and convert them into machine-readable error patterns, which can serve as a starting point for a variety of tasks, includ-

ing explainable automatic error classification and controlled feedback comment generation.

We put a first implementation of this method to the test through an example retrieval task where patterns extracted from a set of example sentence-correction pairs are used to find similar errors in an L1-L2 treebank. An interactive version of the resulting system is also made available as a prototype Computer-Assisted Language Learning (CALL) application, similar to Arai et al. (2019)'s corpus search tool for L2 Japanese learners.<sup>1</sup>

## 2 Related work

Standardizing and automating the annotation of learner corpora is desirable for a variety of purposes. Notable in this sense is ERRANT (Bryant et al., 2017), an automatic ERRor ANnotation Toolkit for learner English whose principal aim is allowing finer-grained evaluation of Grammatical Error Correction (GEC) and Detection (GED) systems. ERRANT extracts edit operations from learner sentence-correction pairs. Each edit is later labelled following an error taxonomy relying solely on dataset-agnostic information such as the POS (Part Of Speech) tag of the tokens involved.

With L1-L2 parallel UD treebanks, there is no explicit error annotation step: the idea is that morphosyntactical annotation should suffice, as error can be described by means of tree patterns pairs, comparing the original learner attempt with its target L1 counterpart (Lee et al., 2017). When it comes to retrieving instances of specific patterns of error, a query engine was developed by Masciolini (2023). Choshen et al. (2020), on the other hand, used UD-annotated parallel data to automatically derive SERCL, a new taxonomy of Syntactic ERRors for automatic CLASSification, later combined with ERRANT's under the name of SERRANT (Choshen et al., 2021). SERCL error types

<sup>1</sup>Our software is available for download at [github.com/harison/L2-UD](https://github.com/harison/L2-UD) (accessed 31.05.2023).



Figure 1: A correct-incorrect UD sentence pair both in English and Swedish, with discrepancies highlighted in bold.

are obtained by concatenating the morphosyntactical features of the head of a problematic text segment before and after correction. The results are labels such as ADJ→ADV (adjective replaced by adverb), applicable for instance to the example in Figure 1. Choshen et al. (2020)’s system, as well as the query tool, has been tested both on manually annotated treebanks and on automatically parsed sentences, with results suggesting the standard parsers’ relative robustness to learner errors.

Querying parallel UD treebanks and using them to automatically derive data-driven error taxonomies are two tasks closely related but not identical to what we attempt in this paper. As opposed to searching for specific error types, we try to detect all errors appearing in an L1-L2 treebank, and rather than classifying them according to a flat labelling scheme we aim at obtaining fine-grained descriptions of each, in the form of patterns meant for further processing.

### 3 Methodology

We see error pattern extraction as a two-stage process. Given a learner sentence and the corresponding correction, the first step, discussed in Section 3.1, is locating its problematic portions to extract error-correction pairs. As per Section 3.2, the latter are then converted into machine-readable patterns.

#### 3.1 Locating error-correction pairs

A simple way to locate errors in a pair of sentences is to phrase- and/or word-align them and consider as erroneous all correspondences presenting any discrepancies between their L1 and L2 components. If the goal is to only select errors belonging to a specific macro-category, the task of deciding whether a discrepant alignment is relevant or not becomes less straightforward. In this case, we are mostly interested in morphosyntax, for which UD annotation is particularly informative. At this stage, however, we assume our data to only contain this type of errors and focus on alignment alone.

That of alignment is a problem common to all the works mentioned in Section 2. To extract edits, ERRANT uses a linguistically-enhanced L1-L2

algorithm (Felice et al., 2016). While reportedly achieving state-of-the-art results, its implementation is English-specific. Choshen et al. (2020), on the other hand, work in a bilingual setting. The paper leaves the details of the alignment step unspecified, but from a superficial inspection of the source code it appears that the same method, along with an *ad-hoc* adaptation to Russian, is used.

Since our aim is to work cross-lingually, we adopt the same approach as Masciolini (2023), consisting in extracting correspondences between UD subtrees using the CONCEPT-ALIGNMENT package (Masciolini and Ranta, 2021). Originally developed for the syntax-based extraction of translation equivalents from multilingual parallel UD treebanks, the library is completely language-agnostic at its core, and its alignment rules can be easily customized to better suit the L1-L2 domain.

Furthermore, extracting subtrees rather than text spans ensures some degree of flexibility in determining how much context to extract for a given error. Depending on the use case, error-correction pairs can consist either of just the tokens involved in the corresponding edit operation, similarly to what is done in SERRANT, or of larger segments, useful to understand why the edit is required. In Figure 1, for instance, both the adverb *slowly* and the adjective *slow* (resp. *långsamt* and *långsam*) are acceptable forms, if taken in isolation: adjectives are only marked as incorrect because they modify a verb. For each detected error, our extraction module produces patterns of various sizes. From the perspective of example retrieval, in fact, smaller patterns are more likely to generate hits, but larger ones result in better matches.

#### 3.2 From CoNLL-U trees to error patterns

Alignments, and therefore errors, are internally represented as pairs of *rose trees*, tree structures with a variable, unbounded number of children per node. While this representation can be easily converted back into CoNLL-U format, which is itself machine-readable, complete UD sentences are too information-rich for most practical purposes and not as easy to manipulate as a recursive data struc-

ture. We therefore describe errors using a UD query language. Among several existing options, we selected the pattern matching language available as part of GF-UD (Kolachina and Ranta, 2016; Ranta and Kolachina, 2017), the easiest to integrate with the rest of the codebase.

**UD patterns** GF-UD essentially provides three types of patterns:<sup>2</sup>

- *single-token patterns*, such as `POS "ADJ"`, matching subtree roots. With a similar syntax, it is possible to pattern match based on the token’s `XPOS`, `DEPREL`, `FEATS`, `FORM` or `LEMMA`, each corresponding a CoNNL-U field<sup>3</sup>;
- *tree patterns* in the form `TREE p [ps]`, where `p` is a pattern to be matched by the root of a subtree and `[ps]` a list of patterns denoting its dependents. `TREE (POS "NOUN") [DEPREL "amod"]`, for instance, matches nouns modified by an adjective;
- *sequence patterns* like `SEQUENCE [DEPREL "amod", POS "NOUN"]`, matching nouns preceded by an adjectival modifier.

In addition, the language allows combining patterns with the logical operators `AND`, `OR` and `NOT` and provides a `TRUE` pattern matching any subtree.

Following Masciolini (2023), we use pairs of these UD patterns to describe the discrepancies between L1 and L2 trees. As a consequence, a way to describe the error in Figure 1 on the basis of POS tags is the following:<sup>4</sup>

```
(TREE_ (POS "VERB") [POS "ADV"],
 TREE_ (POS "VERB") [POS "ADJ"])
```

Here, the first pattern denotes the correct form and the second the erroneous learner attempt. This can be written even more concisely as

```
TREE_ (POS "VERB") [POS {"ADV"→"ADJ"}]
```

This means that, to modify a verb, the learner used an adjective rather than an adverb. If we focus on the edit operation only, we obtain the pattern

```
POS {"ADV"→"ADJ"}
```

equivalent to SERCL/SERRANT’s `ADJ→ADV`.

<sup>2</sup>For the full specification of the GF-UD pattern syntax, see [github.com/GrammaticalFramework/gf-ud/blob/master/doc/patterns.md](https://github.com/GrammaticalFramework/gf-ud/blob/master/doc/patterns.md) (accessed 19.04.2023).

<sup>3</sup>For more information about the UD standard, see [universaldependencies.org](https://universaldependencies.org) (accessed 31.05.2023).

<sup>4</sup>Underscored `TREE_` patterns match even trees having dependents other than those explicitly listed, like Figure 1’s.

Converting alignments to tree pattern pairs, which have the same recursive structure, is extremely simple. The same can be said of sequence patterns, since GF-UD also provides a list-like data type to represent UD sentences and functions to convert between the latter and rose trees. The most straightforward approach, however, yields “full” UD patterns that are excessively specific. For this reason, we develop various simplification strategies producing more general, yet informative patterns.

**Simplification strategies** A first, simple strategy, is to **filter patterns by CoNNL-U field**. This was already exemplified above when only considering Universal POS tags. A less strict options is to take into account all morphosyntactically relevant fields (`FEATS`, `DEPREL`, `POS` and possibly `XPOS`). A way to achieve further simplification is to **remove fields whose values are identical in both components of the patterns**. Another approach is to recursively compare the L1 and L2 sides of an error pattern and **eliminate identical subpatterns**. In addition, it is possible to **simplify single (monolingual) patterns** in various ways, for instance by transforming sequence patterns of length 1 and tree patterns with empty dependent lists into single-token patterns. Appendix A demonstrates the application of these strategies to the example in Figure 1. With example retrieval in mind, we apply all strategies, in sequence, to each extracted pattern, without discarding the intermediate results. This maximizes the chance of finding relevant examples while laying the foundation for ranking the results.

## 4 Preliminary evaluation

We carry out a first evaluation of our method through an example retrieval task. In particular, we try to find occurrences of errors similar to those extracted from a given sentence-correction pair in an L1-L2 treebank. Implementation-wise, this is done by combining our error extraction module with Masciolini (2023)’s query engine: run on an input pair, the extraction procedure returns one or more patterns, in turn used to query the treebank.

We make an interactive version of such error retrieval pipeline also available as a prototype CALL application, analogous to the incorrect example retrieval tool presented in Arai et al. (2019). In this case, input sentences are entered as text and parsed on the fly using UDPipe’s REST API.<sup>5</sup>

<sup>5</sup>[lindat.mff.cuni.cz/services/udpipe/api-reference.php](https://lindat.mff.cuni.cz/services/udpipe/api-reference.php) (accessed 31.05.2023).

## 4.1 Data

While the final iteration of our extraction method will be meant for authentic learner data, we carry out this first evaluation on two datasets for linguistic acceptability judgments composed of minimal correct-incorrect sentence pairs isolating specific linguistic phenomena, i.e. where the incorrect element contains a single grammatical error. In this way, we postpone dealing with the complexities that can arise from the simultaneous presence of several errors involving the same tokens. We simplify the task further by filtering out sentences containing errors beyond mere morphosyntax, such as incorrect lexical choices and spelling mistakes, for which automatic UD annotation is less informative and potentially misleading.

**BLIMP** The Benchmark of Linguistic Minimal Pairs (BLIMP) (Warstadt et al., 2020), developed for evaluating the linguistic knowledge of language models, is a dataset consisting of 67 subsets, each containing 1 000 correct-incorrect sentence pairs exemplifying a specific error type or *paradigm*. Examples are artificially generated based on linguist-crafted templates and subsets are organized in 12 groups on the basis of the linguistic phenomenon they describe. Based on their metadata, we select lexically identical pairs marked as belonging to the fields of morphology or syntax and parse them with UDPipe 2 (Straka, 2018)’s default English model. The result is a parallel treebank of 14 996 sentences, 100 of which we set aside as inputs for the example retrieval pipeline. Specifically, we extract patterns from this 100-sentence subset and match them against the remaining 14 896 pairs to retrieve similar correct-incorrect examples.<sup>6</sup>

**DALAJ** The Dataset for Linguistic Acceptability Judgments (DALAJ) is, in turn, composed of L2 Swedish sentence-correction minimal pairs derived from the error-annotated SWELL Swedish Language Learner corpus (Volodina et al., 2019) and therefore arguably closer to the data our system is being built for.<sup>7</sup> SWELL uses a two-level error taxonomy: labels, such as M-Adj/adv, are composed of a capital letter, indicating the error’s macro-category (in this case, Morphology), followed by an abbreviation

<sup>6</sup>The BLIMP splits used in this paper, as well as the preprocessing scripts, are available at [github.com/harison/L1-L2-BLIMP/tree/bea](https://github.com/harison/L1-L2-BLIMP/tree/bea) (accessed 31.05.2023).

<sup>7</sup>An early version of DALAJ, covering only lexical errors, is presented in Volodina et al. (2021).

specifying the affected POS and/or morphological features. The M-Adj/adv label, for instance, refers to Adjective forms corrected with the corresponding adverb, such as *långsamt* → *långsam\** in the example displayed in Figure 1. We select the 1 198 error-correction pairs belonging to the M and S macro-categories and process them analogously to BLIMP data, the only difference being the usage of a Swedish model.<sup>8</sup>

## 4.2 Results

Ideally, quantitatively evaluating the performance of our system on the example retrieval task defined above would involve computing the precision and recall of each query performed with the extracted patterns. In practice, however, this is unfeasible in our current setup, as it would require manually inspecting all matches. While an identity of error labels between the input pair and a match is generally a good indication of a true positive, in fact, it is not at all always the case that different labels correspond to a false positive: the same error can sometimes be interpreted, and therefore labelled, differently. The Swedish word *långsamt*, for instance, is both an adverb (“slowly”) and the singular neuter form of the adjective *långsam* (“slow”), meaning that a phrase like *ett {långsamt → långsam\*} tempo* (“a slow tempo”, where *{långsamt → långsam\*}* modifies the neuter noun *tempo*) could, following the SWELL annotation guidelines (Rudebeck and Sundberg, 2021), be annotated both as M-Adj/adv and M-Gend. For similar reasons, counting actual false negatives is also challenging.

Instead, for each dataset, we compute the retrieval rate  $R$ , i.e. the percentage of sentences for which the system was able to return one or more matches, regardless of their correctness, and compare it with the *successful retrieval rate*  $R_+$ , where only sentences with at least one relevant match was found. Since we use search results as a proxy of the usefulness of the extracted patterns rather than to assess the performance of the query engine, we deem this to be sufficient for a first evaluation. Results are summarized in the table below.

	BLIMP	DALAJ
$R$	82%	69%
$R_+$	82%	63%

<sup>8</sup>The DALAJ splits used in this paper, as well as the preprocessing scripts, are available at [github.com/harison/L1-L2-DaLAJ/tree/bea](https://github.com/harison/L1-L2-DaLAJ/tree/bea) (accessed 31.05.2023).

Figures for BLIMP, whose data is controlled and finely categorized by paradigm, were obtained fully automatically by checking whether one or more of the retrieved examples belonged to the same subset. DALAJ matches, on the other hand, still required manual inspection due to the dataset's coarser-grained labelling scheme and the scarcer predictability of the sentences. More specifically, we checked the search results of each query looking for relevant matches, defined, for the sake of this evaluation, as examples presenting an error similar to that of the input pair, regardless of the degree of specificity and granularity of the extracted pattern(s). Given the input *de blev {utsatta → utsattad\*} på två olika sätt* ("they were exposed in two different ways", where the adjective *utsattad\**, "exposed", is incorrectly inflected for number), for instance, this implied considering the sentences *{promenader → promenad\*} är bra för människors hälsa* ("{walks → walk\*} are good for people's health", where the number inflection error involves the noun) and *vi är {glada → glad\*} varje dag* ("we are happy every day", where the incorrectly inflected word is again an adjective, *glad*) even though only the latter involves the same POS<sup>9</sup>. While results are encouraging for both datasets, we observe a marked difference between the two in terms of retrieval rate. Several different factors might contribute to this: the difference in size between the two corpora, the fact that all pairs we selected from BLIMP, but not from DALAJ, are lexically identical and some intrinsic characteristics of the BLIMP dataset, such as the template-based method used to generate its sentences.

In cases where no or exclusively incorrect matches are found, failures may also be caused by parse errors, issues related with the query engine or, especially when it comes to the smaller Swedish treebank, merely by a lack of similar examples in the corpus. In such instances, we investigate further by inspecting the UD trees and extracted patterns. When it comes to BLIMP data, pairs with no matches belong in all but one case to the island effects group, comprising word order errors related to *wh*-words, such as *Whose {hat should Tonya wear → should Tonya wear hat\*}?* Unsurprisingly, errors of this kind pose a challenge for the parser and therefore often incorrectly aligned.

Word order errors are problematic in Swedish

<sup>9</sup>See Appendix B for a similar example, where the same sentence matches two patterns of different sizes.

too, but even other syntactical errors, most notably S-Clause (change of basic clause structure), S-MSubj (missing subject) and M-Adj/adv<sup>10</sup> (adjective corrected to adverb form, as in Figure 1) appear to cause issues at the parsing stage, especially when corrections involve complex rephrasings and/or lexical changes. Morphological errors involving nonexistent word forms are also often handled incorrectly. An example of that is the Swedish L2 sentence *Kommunikationen hade dittills skett via brev, och brevutdelning fick man fem {gångar → gångar\*} om dagen* ("Communication had until then taken place by mail, and letters were delivered five times a day"), where *gångar* is an incorrect plural form of the noun *gång*, corrected to *gångar*. In such cases, the morphological analysis of L2 is identical to that of the L1 and the only usable patterns are those preserving lexical information, for which finding treebank matches is less likely.

## 5 Conclusions and future work

We presented a novel approach for extracting morphosyntactical error patterns from L1-L2 parallel UD treebanks and put it to the test through an example retrieval task. While performed on datasets for linguistic acceptability judgments rather than authentic learner data, our preliminary evaluation gave promising results and provided helpful insights for the further development of the tool.

Future work on the extraction method itself will focus on handling nonexistent word forms and dealing with the complexity of actual L2 data. Real-world L2 texts come with two main challenges: handling non-morphosyntactical errors, such as spelling mistakes and incorrect lexical choices, and isolating each of the grammatical errors occurring in the same sentence. We mentioned that our system extracts patterns of different sizes and at varying degrees of simplification, whose usefulness depends on the use case. This drives us to also investigate pattern selection and ranking. The latter, together with a more user-friendly interface, could contribute to the improvement the example retrieval pipeline to better suit the learners' needs. Further improvements will require addressing the L2 parsing issues identified through our preliminary evaluation, for instance by fine-tuning a UDpipe model on L2 data, and possibly intervening on the alignment step.

<sup>10</sup>Even though SWELL classifies this as a morphological error, it is syntactical from a UD perspective.

## References

- Mio Arai, Masahiro Kaneko, and Mamoru Komachi. 2019. [Grammatical-error-aware incorrect example retrieval system for learners of Japanese as a second language](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 296–305, Florence, Italy. Association for Computational Linguistics.
- Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. [Automatic annotation and evaluation of error types for grammatical error correction](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.
- Leshem Choshen, Dmitry Nikolaev, Yevgeni Berzak, and Omri Abend. 2020. [Classifying syntactic errors in learner language](#). In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 97–107, Online. Association for Computational Linguistics.
- Leshem Choshen, Matanel Oren, Dmitry Nikolaev, and Omri Abend. 2021. SERRANT: a syntactic classifier for english grammatical error types. *arXiv preprint arXiv:2104.02310*.
- Mariano Felice, Christopher Bryant, and Ted Briscoe. 2016. [Automatic extraction of learner errors in ESL sentences using linguistically enhanced alignments](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 825–835, Osaka, Japan. The COLING 2016 Organizing Committee.
- Prasanth Kolachina and Aarne Ranta. 2016. [From abstract syntax to Universal Dependencies](#). volume 13. CSLI Publications.
- John Lee, Keying Li, and Herman Leung. 2017. [L1-L2 parallel dependency treebank as learner corpus](#). In *Proceedings of the 15th International Conference on Parsing Technologies*, pages 44–49, Pisa, Italy. Association for Computational Linguistics.
- Arianna Masciolini. 2023. [A query engine for L1-L2 parallel dependency treebanks](#). In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 574–587, Tórshavn, Faroe Islands. University of Tartu Library.
- Arianna Masciolini and Aarne Ranta. 2021. [Grammar-based concept alignment for domain-specific Machine Translation](#). In *Proceedings of the Seventh International Workshop on Controlled Natural Language (CNL 2020/21)*, Amsterdam, Netherlands. Special Interest Group on Controlled Natural Language.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. [Universal Dependencies v2: An evergrowing multilingual treebank collection](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.
- Aarne Ranta and Prasanth Kolachina. 2017. [From Universal Dependencies to abstract syntax](#). In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 107–116, Gothenburg, Sweden. Association for Computational Linguistics.
- Lisa Rudebeck and Gunlög Sundberg. 2021. SweLL correction annotation guidelines. In *The SweLL guideline series nr 4*, Gothenburg, Sweden. Institutionen för svenska, Göteborgs Universitet.
- Milan Straka. 2018. [UDPipe 2.0 prototype at CoNLL 2018 UD shared task](#). In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.
- Elena Volodina, Lena Granstedt, Arild Matsson, Beáta Megyesi, Ildikó Pilán, Julia Prentice, Dan Rosén, Lisa Rudebeck, Carl-Johan Schenström, Gunlög Sundberg, et al. 2019. The SweLL language learner corpus: From design to annotation. *Northern European Journal of Language Technology*, 6:67–104.
- Elena Volodina, Yousuf Ali Mohammed, and Julia Klezl. 2021. DaLAJ-a dataset for linguistic acceptability judgments for Swedish: Format, baseline, sharing. *arXiv preprint arXiv:2105.06681*.
- Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. [BLiMP: The benchmark of linguistic minimal pairs for English](#). *Transactions of the Association for Computational Linguistics*, 8:377–392.

## A Application of simplification strategies

Input correct-incorrect sentence pair: *<I write slowly, I write slow>*.

### 0. largest complete extracted error pattern:

```
TREE
  (AND [
    FORM "write",
    LEMMA "write",
    POS "VERB",
    XPOS "VBP",
    FEATS "Mood=Ind|Number=Sing|Person=1|Tense=Pres|VerbForm=Fin",
    DEPREL "root"])
  [AND [
    FORM "I",
    LEMMA "I",
    POS "PRON",
    XPOS "PRP",
    FEATS "Case=Nom|Number=Sing|Person=1|PronType=Prs",
    DEPREL "nsubj"],
  AND [
    FORM {"slowly" → "slow"},
    LEMMA {"slowly" → "slow"},
    POS {"ADV" → "ADJ"},
    XPOS {"RB" → "JJ"},
    FEATS "_",
    DEPREL {"advmod" → "amod"}]]
```

### 1. filtering by CoNNL-U field, keeping only morphosyntax-related fields (UPOS, FEATS and DEPREL):

```
TREE
  (AND [
    POS "VERB",
    FEATS "Mood=Ind|Number=Sing|Person=1|Tense=Pres|VerbForm=Fin",
    DEPREL "root"])
  [AND [
    POS "PRON",
    FEATS "Case=Nom|Number=Sing|Person=1|PronType=Prs",
    DEPREL "nsubj"],
  AND [
    POS {"ADV" → "ADJ"},
    FEATS "_",
    DEPREL {"advmod" → "amod"}]]
```

### 2. removal of fields whose values are identical everywhere in both the L1 and L2 component:

```
TREE
  (AND [POS "VERB", DEPREL "root"])
  [AND [POS "PRON", DEPREL "nsubj"],
  AND [POS {"ADV" → "ADJ"}, DEPREL {"advmod" → "amod"}]]
```

### 3. elimination of identical subpatterns:

```
TREE (TRUE) [TRUE, AND [POS {"ADV" → "ADJ"}, DEPREL {"advmod" → "amod"}]]
```

### 4. monolingual single-pattern simplifications:

```
AND [POS {"ADV" → "ADJ"}, DEPREL {"advmod" → "amod"}]]
```

## B Example program output<sup>11</sup>

Input correct-incorrect sentence pair: (*jag skriver långsamt, jag skriver långsam*).

### Sentence 391

L1 sentence

För det andra kommer studenterna ibland så **tidigt** så de måste vänta i en korridor istället för att vänta på ett café och dricka kaffe eller te .

L2 sentence

För det andra kommer studenterna ibland så **tidig** så de måste vänta i en korridor istället för att vänta på ett café och dricka kaffe eller .

### Sentence 395

L1 sentence

När man inte har någon bil , får man promenera till jobbet eller ta bussen ; Det går inte så **snabbt** , och man måste planera lite mer , men det är naturligt för oss .

L2 sentence

När man inte har någon bil , får man promenera till jobbet eller ta bussen ; Det går inte så **snabb** , och man måste planera lite mer , men det är naturligt för oss .

### Sentence 684

L1 sentence

Och just nu känns vårt liv **jättebra** .  
Och just nu **känns** vårt liv **jättebra** .

L2 sentence

Och just nu känns våras liv **jättebra** .  
Och just nu **känns** våras liv **jättebra** .

### Sentence 459

L1 sentence

På senare år har engelskan kommit att få en allt starkare ställning **internationellt** och också i Sverige .  
På senare år har engelskan kommit att **få** en allt starkare ställning **internationellt** och också i Sverige .

L2 sentence

På senare år har engelskan kommit att få en allt starkare ställning **internationell** och också i Sverige .  
På senare år har engelskan kommit att **få** en allt starkare ställning **internationell** och också i Sverige .

### Sentence 436

L1 sentence

Jag är väldigt glad över det eftersom jag tycker att det finns för många människor , **speciellt** barn , som ser kläder som en statussymbol och köper dem även om de har inte tillräckligt med pengar .

L2 sentence

Jag är väldigt glad över det eftersom jag tycker att det finns för många människor , **speciell** barn , som ser kläder som en statussymbol och köper dem även om de har inte tillräckligt med pengar .

<sup>11</sup>Results obtained on the DALAJ treebank with the latest version of the interactive example retrieval pipeline (example command of L2-UD, run with the `-markdown` option), with commit SHA 9a1ec851313a4c3176826c77aa677e94158c3519. As it is to be expected, some sentences match several of the extracted patterns. While seemingly identical matches have been manually removed for the sake of compactness, highlighting clearly shows that sentences like 459 match not only the single-token POS { "ADV" → "ADJ" } pattern, but also the more specific TREE\_ (POS "VERB") [POS { "ADV" → "ADJ" } ] pattern and could therefore be ranked higher.



### Sentence 1017

L1 sentence

Men i Sverige går det **bättre** för bönderna !

L2 sentence

Men i Sverige går det **bästa** för bönderna !

### Sentence 437

L1 sentence

Om man skulle välja att gå emot normen så skulle det leda till utanförskap , vilket är någonting jag inte tror att någon vill uppleva , och därför väljer jag att klä mig **likadant** som de andra på mitt jobb .  
Om man skulle välja att gå emot normen så skulle det leda till utanförskap , vilket är någonting jag inte tror att någon vill uppleva , och därför väljer jag att **klä** mig **likadant** som de andra på mitt jobb .

L2 sentence

Om man skulle välja att gå emot normen så skulle det leda till utanförskap , vilket är någonting jag inte tror att någon vill uppleva , och därför väljer jag att klä mig **likadan** som de andra på mitt jobb .  
Om man skulle välja att gå emot normen så skulle det leda till utanförskap , vilket är någonting jag inte tror att någon vill uppleva , och därför väljer jag att **klä** mig **likadan** som de andra på mitt jobb .

### Sentence 420

L1 sentence

Det finns **säkert** en del som undrar varför de finska ungdomarna obligatoriskt ska läsa svenska i finska skolor när endast cirka sex procent av befolkningen läser svenska som modersmål .  
Det **finns säkert** en del som undrar varför de finska ungdomarna obligatoriskt ska läsa svenska i finska skolor när endast cirka sex procent av befolkningen läser svenska som modersmål .

L2 sentence

Det finns **säker** en del som undrar varför de finska ungdomarna obligatoriskt ska läsa svenska i finska skolor när endast cirka sex procent av befolkningen läser svenska som modersmål .  
Det **finns säker** en del som undrar varför de finska ungdomarna obligatoriskt ska läsa svenska i finska skolor när endast cirka sex procent av befolkningen läser svenska som modersmål .

### Sentence 407

L1 sentence

Andra punkten : Vi behöver biblioteket för att där finns böcker på olika språk , **specifikt** mitt modersmål .

L2 sentence

Andra punkten : Vi behöver biblioteket för att där finns böcker på olika språk , **specifik** mitt modersmål .

### Sentence 392

L1 sentence

Det är viktigt för mig när jag behöver ta det lite **lugnt** och göra mina läxor , och det är viktigt för mig att prata svenska med en svensk person och lära mig många nya ord .

L2 sentence

Det är viktigt för mig när jag behöver ta det lite **lugna** och göra mina läxor , och det är viktigt för mig att prata svenska med en svensk person och lära mig många nya ord .

### Sentence 425

L1 sentence

Historier som från början bara var **munligt** berättade tar idag alla tänkbara former och förekommer som musik , teater , romaner , serier , filmer och spel .

L2 sentence

Historier som från början bara var **munlig** berättade tar idag alla tänkbara former och förekommer som musik , teater , romaner , serier , filmer och spel .

**Sentence 429**

L1 sentence

I boken ” Stjärnlösa nätter ” så ser man tydligt hur en hatkärlek kan påverka en människas liv både **negativt** och positivt .

L2 sentence

I boken ” Stjärnlösa nätter ” så ser man tydligt hur en hatkärlek kan påverka en människas liv både **negativ** och positivt .

**Sentence 984**

L1 sentence

Där sitter jag med min familj och äter , sjunger , dansar , skrattar , leker och studerar ... I hemmet kommer jag **jättenära** min son och jag kan lära honom mycket om livet och **hur** han kan bli bra person .

L2 sentence

Där sitter jag med min familj och äter , sjunger , dansar , skrattar , leker och studerar ... I hemmet kommer jag **jättenärmare** min son och jag kan lära honom mycket om livet och hur han kan bli bra person .

**Sentence 401**

L1 sentence

Jag tycker att buss är bättre än bil eftersom det är lättare att använda buss än bil , för alla människor , **särskilt** de fattiga , kan använda buss som de vill .

L2 sentence

Jag tycker att buss är bättre än bil eftersom det är lättare att använda buss än bil , för alla människor , **särskild** de fattiga , kan använda buss som de vill .

**Sentence 457**

L1 sentence

Jag lärde mig att om saker inte går bra för dig ska du vara modig och ta det **lugnt** , det kommer att bli bättre , ge bara aldrig upp !  
Jag lärde mig att om saker inte går bra för dig ska du vara modig och **ta** det **lugnt** , det kommer att bli bättre , ge bara aldrig upp !

L2 sentence

Jag lärde mig att om saker inte går bra för dig ska du vara modig och ta det **lugn** , det kommer att bli bättre , ge bara aldrig upp !  
Jag lärde mig att om saker inte går bra för dig ska du vara modig och **ta** det **lugn** , det kommer att bli bättre , ge bara aldrig upp !

**Sentence 442**

L1 sentence

Det finns olika sätt som man kan använda eller uttrycka sig på för att kunna kommunicera med varandra , till exempel skrivet eller **munligt** med hjälp av ord på en mängd olika språk .

L2 sentence

Det finns olika sätt som man kan använda eller uttrycka sig på för att kunna kommunicera med varandra , till exempel skrivet eller **munlig** med hjälp av ord på en mängd olika språk .

**Sentence 421**

L1 sentence

Detta leder till motstånd från landets folk som ser **negativt** på regeringens maktfullkomliga metod .  
Detta leder till motstånd från landets folk som ser **negativt** på regeringens maktfullkomliga metod .

L2 sentence

Detta leder till motstånd från landets folk som ser **negativ** på regeringens maktfullkomliga metod .  
Detta leder till motstånd från landets folk som ser **negativ** på regeringens maktfullkomliga metod .

**Sentence 431**

L1 sentence

Historier som från början bara var **muntligt** berättade tar idag alla tänkbara former och förekommer som musik , teater , poesi , romaner , serier , filmer och spel .

L2 sentence

Historier som från början bara var **muntliga** berättade tar idag alla tänkbara former och förekommer som musik , teater , poesi , romaner , serier , filmer och spel .

**Sentence 458**

L1 sentence

Det är inte så lätt att svara **snabbt** .  
Det är inte så lätt att **svara snabbt** .

L2 sentence

Det är inte så lätt att svara **snabb** .  
Det är inte så lätt att **svara snabb** .

**Sentence 451**

L1 sentence

Mitt råd är att du måste ta det **lugnt** och fokusera , till exempel klä på dig fina kläder , det betyder inte smustiga kläder , eller du kan använda parfym , men inte så mycket .  
Mitt råd är att du måste **ta** det **lugnt** och fokusera , till exempel klä på dig fina kläder , det betyder inte smustiga kläder , eller du kan använda parfym , men inte så mycket .

L2 sentence

Mitt råd är att du måste ta det **lugn** och fokusera , till exempel klä på dig fina kläder , det betyder inte smustiga kläder , eller du kan använda parfym , men inte så mycket .  
Mitt råd är att du måste **ta** det **lugn** och fokusera , till exempel klä på dig fina kläder , det betyder inte smustiga kläder , eller du kan använda parfym , men inte så mycket .

**Sentence 466**

L1 sentence

Jag personligen lägger inte **medvetet** så stor vikt vid kläder , kanske för att den miljö som jag lever i eller de människor som jag umgås med inte ser kläder som något betydelsefullt .  
Jag personligen **lägger** inte **medvetet** så stor vikt vid kläder , kanske för att den miljö som jag lever i eller de människor som jag umgås med inte ser kläder som något betydelsefullt .

L2 sentence

Jag personligen lägger inte **medveten** så stor vikt vid kläder , kanske för att den miljö som jag lever i eller de människor som jag umgås med inte ser kläder som något betydelsefullt .  
Jag personligen **lägger** inte **medveten** så stor vikt vid kläder , kanske för att den miljö som jag lever i eller de människor som jag umgås med inte ser kläder som något betydelsefullt .

**Sentence 462**

L1 sentence

Alla mina dagar gick så **dåligt** .

L2 sentence

Alla mina dagar gick så **dålig** .

### Sentence 461

L1 sentence

**Sammanfattat** har jag en föränderlig relation till kläder , men det viktigaste är att de möjliggör allt jag vill uppleva , från bergsvandring till fest .  
**Sammanfattat har** jag en föränderlig relation till kläder , men det viktigaste är att de möjliggör allt jag vill uppleva , från bergsvandring till fest .

L2 sentence

**Sammanfattad** har jag en föränderlig relation till kläder , men det viktigaste är att de möjliggör allt jag vill uppleva , från bergsvandring till fest .  
**Sammanfattad har** jag en föränderlig relation till kläder , men det viktigaste är att de möjliggör allt jag vill uppleva , från bergsvandring till fest .

### Sentence 390

L1 sentence

Dessutom är det **troligen** kö då alla vill ha rast och kaffe samtidigt .

L2 sentence

Dessutom är det **troliget** kö då alla vill ha rast och kaffe samtidigt .

### Sentence 387

L1 sentence

Det var ganska svårt **först** men jag är van och lärde mig själv hur man bor och anpassar sig i ett nytt land .

L2 sentence

Det var ganska svårt **första** men jag är van och lärde mig själv hur man bor och anpassar sig i ett nytt land .

### Sentence 469

L1 sentence

Tänk **positivt** istället så kommer du att hitta många betydelsefulla saker inom din familj .  
**Tänk positivt** istället så kommer du att hitta många betydelsefulla saker inom din familj .

L2 sentence

Tänk **positiv** istället så kommer du att hitta många betydelsefulla saker inom din familj .  
**Tänk positiv** istället så kommer du att hitta många betydelsefulla saker inom din familj .

### Sentence 467

L1 sentence

Detta kan dock skapa svårigheter med att kunna förbereda och undervisa ungdomar **tillräckligt** .  
Detta kan dock skapa svårigheter med att kunna **förbereda** och undervisa ungdomar **tillräckligt** .

L2 sentence

Detta kan dock skapa svårigheter med att kunna förbereda och undervisa ungdomar **tillräckliga** .  
Detta kan dock skapa svårigheter med att kunna **förbereda** och undervisa ungdomar **tillräckliga** .

### Sentence 410

L1 sentence

Efter några år visade inspektörerna rapporter om att det nog fanns lite kokain i coca cola , men tyvärr ville de inte kommunicera detta **offentligt** .  
Efter några år visade inspektörerna rapporter om att det nog fanns lite kokain i coca cola , men tyvärr ville de inte **kommunicera** detta **offentligt** .

L2 sentence

Efter några år visade inspektörerna rapporter om att det nog fanns lite kokain i coca cola , men tyvärr ville de inte kommunicera detta **offentlig** .  
Efter några år visade inspektörerna rapporter om att det nog fanns lite kokain i coca cola , men tyvärr ville de inte **kommunicera** detta **offentlig** .

### Sentence 375

L1 sentence

Hon lär ut svenska mycket **snällt** och fint .

L2 sentence

Hon lär ut svenska mycket **snäll** och fint .

### Sentence 463

L1 sentence

Jag hoppas kunna lära mig **snabbt** och börja söka jobb .

Jag hoppas kunna **lära** mig **snabbt** och börja söka jobb .

L2 sentence

Jag hoppas kunna lära mig **snabb** och börja söka jobb .

Jag hoppas kunna **lära** mig **snabb** och börja söka jobb .

# Learning from Partially Annotated Data: Example-aware Creation of Gap-filling Exercises for Language Learning

Semere Kiros Bitew\*, Johannes Deleu\*, A. Seza Dođruöz, Chris Develder  
and Thomas Demeester

IDLab, Ghent University - imec

\* Equal contribution

{semerekiros.bitew, as.dogruoz, firstname.lastname}@ugent.be

## Abstract

Since performing exercises (including, e.g., practice tests) forms a crucial component of learning, and creating such exercises requires non-trivial effort from the teacher. There is a great value in automatic exercise generation in digital tools in education. In this paper, we particularly focus on automatic creation of gap-filling exercises for language learning, specifically grammar exercises. Since providing any annotation in this domain requires human expert effort, we aim to avoid it entirely and explore the task of converting existing texts into new gap-filling exercises, purely based on an example exercise, *without explicit instruction or detailed annotation* of the intended grammar topics. We contribute (i) a novel neural network architecture specifically designed for aforementioned gap-filling exercise generation task, and (ii) a real-world benchmark dataset for French grammar. We show that our model for this French grammar gap-filling exercise generation outperforms a competitive baseline classifier by 8% in F1 percentage points, achieving an average F1 score of 82%. Our model implementation and the dataset are made publicly available<sup>1</sup> to foster future research, thus offering a standardized evaluation and baseline solution of the proposed partially annotated data prediction task in grammar exercise creation.

## 1 Introduction

While digital education tools have been increasingly developed and deployed for over a decade, the e-learning sector has definitely boomed in the wake of COVID-19, even leading to a new Digital Education Action Plan from the European Commission.<sup>2</sup> As one application in e-learning, we particularly focus on language education, and specifically on the automatic generation of gap-filling grammar exercises. This type of exercises has been

shown to be very effective in language learning, with a noticeable effect of such practice tests on students progress and is generally considered as a global measure of language proficiency (Oller Jr, 1973). Furthermore, automatic generation of exercises has been shown produce relatively high quality exercises, for example, for multiple choice questions (Mitkov et al., 2006), demonstrating the potential effectiveness of reducing human effort and offering cost-effective solutions towards personalized exercise generation. In terms of technology, recent developments in natural language processing, e.g., BERT (Devlin et al., 2018), GPT-3 (Brown et al., 2020), InstructGPT (Ouyang et al., 2022), open up new opportunities for further up-scaling and improving automatic generation of such tests/exercises.

In this paper we specifically propose to generate grammar exercises from existing texts, by inducing well-chosen gaps in a given input sentence, following a set of given example exercise sentences. Further, we aim to create models that can be trained on the exercises themselves, without further annotations. The latter implies that we want to forgo a fully supervised learning setting, because such models would require each gap in the available exercises to be manually annotated with additional metadata, such as the particular exercise type, e.g., for gap-filling exercises, a suitable category such as a verb tense. Thus, we focus on converting given input texts into gap-filling exercises, by mimicking the implicit rules underlying a given example exercise, rather than by following explicit instructions such as a prescribed exercise type.

**Application scenario:** Consider a language teacher, who just introduced a particular grammatical topic (e.g., a new verb tense), and needs the students to practice. The grammar topic of interest may need to be practiced in combination with particular other topics (e.g., related tenses already stud-

<sup>1</sup><https://github.com/semerekiros/GF2/>

<sup>2</sup><https://education.ec.europa.eu/focus-topics/digital-education/action-plan>

ied by the students). Given that gap-filling questions can be completed online and automatically assessed (Daradoumis et al., 2019), the teacher creates a new gap-filling exercise, covering these combined grammar topics. The goal of our model is then to support the automatic creation of new exercises, based on that example exercise, by transforming other texts provided by the teacher into additional gap-filling exercises that target the same linguistic topics to be practiced, without explicit instructions by the teacher of which topics the model should include. This would allow the teacher to rapidly create new training material for the students, potentially more diverse, for example, in terms of topics of the texts, their temporal relevance, or the inherent linguistic difficulty.

**Learning from partially annotated data:** The scenario outlined above represents a learning task in between one-shot learning (i.e., learning from one example (Wang et al., 2020) and full supervision (i.e., based on the full annotation of all examples). On the one hand, the one-shot setting considers the example exercise as a single training instance defining the nature of the prediction task by the way it was constructed by the teacher (in this case, the included grammar topics). On the other hand, the fully supervised setting would require at least explicit knowledge of all exercise instructions (i.e., gap types per exercise). Although we assume the availability of an entire corpus of such exercises, on overlapping grammar topics, we will not rely on explicit annotation of the nature of the gaps (i.e., gap type that defines the type/scope of the grammar exercise, or even just identifying the word category). Thus, we do want to learn from partially annotated examples, where the annotation is limited to just the indication of the gap and the text span that constitutes the expected answer. This basically amounts to the type of information that would be available in a one-/few-shot setting, but we aim to leverage the complete corpus to train our models.

Note that, while creating exercises, teachers are aware of the envisioned exercise type and the gap types, and such exercise type would also be communicated (e.g., as a free-text instruction) to students. Still, to keep our experiments and the gained insights transparent, we left out any exercise level instructions for our experiments.

**Link with related research:** In broad terms, the proposed work fits within the area of automatic question generation (AQG) for the educational domain. In the field of education, creating questions manually is an arduous task that demands considerable time, training, experience, and resources from educators (Davis, 2009). As a solution to this challenge, researchers have turned towards AGQ approaches to automatically generate homework, test, and exam exercises from readily available plain text that requires little to no human calibration. In particular, educational AQG systems have been developed for generating *factoid questions* covering several subjects such as history (Al-Yahya, 2011; Papasalouros et al., 2008), general sciences (Sun et al., 2018; Stasaski and Hearst, 2017; Conejo et al., 2016), health and biomedical sciences (Pugh et al., 2016; Afzal and Mitkov, 2014), etc., as well as for *language learning* such as vocabulary or grammar exercises (Susanti et al., 2017; Hill and Simha, 2016; Goto et al., 2010). There has been some more generic recent work, however, on finding distractors for multiple choice questions across subjects and languages (Bitew et al., 2022). It is in line with recent work on training deep neural networks for general-purpose question generation (Du et al., 2017), based on large training sets. There is a clear preference for two question types that allow for automated assessment, i.e., multiple-choice questions (e.g., in (Stasaski and Hearst, 2017; Pugh et al., 2016; Afzal and Mitkov, 2014; Papasalouros et al., 2008)) or gap-filling questions (as in (Hill and Simha, 2016; Malinova and Rahneva, 2016; Perez-Beltrachini et al., 2012; Goto et al., 2010)).

Our work is focused on gap-filling questions, which typically require test-takers to fill in blank spaces in a text with missing word(s) omitted by test developers. The missing words can either be chosen from a set of possible answers (i.e., closed cloze questions), or generated from scratch using hints provided in the text (i.e., open cloze questions). To generate such questions, various strategies were employed, such as deleting every *n*th word from a text (Taylor, 1953), or rationally deleting words according to specific purpose, e.g., usage of prepositions (Lee and Seneff, 2007), verbs (Sumita et al., 2005) etc. Previous studies have relied on selecting informative sentences (Slavuj et al., 2021; Pino et al., 2008) from existing corpora, such as textbooks (Agarwal and Mannem, 2011), WordNet (Pino et al., 2008), and

### Example 1

- 1 Vous travaillerez beaucoup?  
1 Will you work a lot?
- 2 En ne mangeant plus de bonbons, tu maigriras vite!  
2 By not eating sweets, you will lose weight quickly!
- 3 J'espère que mon équipe favorite ne perdra plus aucun match.  
3 I hope my favorite team won't lose any more games.
- 4 Maxime m'a promis qu'il ne mentira plus jamais.  
4 Maxime promised me that he will never lie again.
- 5 Maman préparera des spaghettis ce soir.  
5 Mum will make spaghetti tonight.

### Example 2

A l'âge de 27 ans, le Californien David Blancarte  
At the age of 27, Californian David Blancarte had  
a eu un grave accident de scooter. Quand il s'est  
a serious scooter accident. When he woke up  
réveillé à l'hôpital, il ne sentait plus ses  
in the hospital, he no longer felt his  
jambes. On lui a expliqué qu'il ne pourrait  
legs. It was explained to him that he couldn't  
plus marcher. C' était une vraie catastrophe  
walk anymore. It was a real disaster for him!  
pour lui! Pendant une longue période de  
During a long period of rehabilitation, he learned  
revalidation, il a appris à se déplacer en chaise.  
to move around in a wheelchair.  
roulante. ...

Figure 1: French grammar exercise from the GF2 corpus, with English translations for convenience shown in light grey. Green spans (with solid underline) are actual gaps as selected by teachers in the dataset, red spans represent potential gaps on other grammar topics but were not marked as gaps. (Left) Isolated sentence exercise with focus on a single tense (*futur simple*); (right) full text exercise combining two tense types (*imparfait* and *passé composé*).

then using techniques such as POS tagging (Agarwal and Mannem, 2011) or term frequency analysis (Mitkov et al., 2006) to determine gap positions. More recently, Marrese-Taylor et al. (2018), have developed sequence labeling model to automate the process of generating gap-filling exercises.

Another very relevant work by Felice et al. (2022) devised a method to adapt an ELECTRA (Clark et al., 2020) model for the purpose of generating open cloze grammar exercises in English. Their approach involved classifying each individual token as either a gap or non-gap. However, there exist several notable distinctions between their approach and our own. Firstly, unlike their method that solely focused on individual tokens, we make gap decisions based on spans. This distinction is essential as our gaps can encompass multiple words, allowing for more comprehensive and contextually accurate grammar exercises. Secondly, our objective and experimental setup differ significantly. Our ultimate goal is to generate multiple versions of the same text, with each version targeting a distinct grammar aspect (e.g., future tense, prepositions of time or combinations of different

types). In contrast, their approach consistently produces exercises of the same type for a given input text (i.e., similar to our baseline model), lacking the versatility and adaptability our model offers.

We observed a tendency in generation of gap-filling questions aiming at well-defined tasks. To the best of our knowledge, none of the prior works have proposed strategies to capture common underlying structures in terms of task definition, while training on a heterogeneous set of real-world examples (e.g., covering various grammatical topics).

#### Key research contributions:

- We introduce the task of the example-aware prediction of suitable linguistic gaps in texts based on partially annotated data. This task is of paramount importance in the development of new gap-filling exercises.
- We present our real-world dataset of French gap-filling exercises covering unknown combinations of grammatical aspects. Our dataset called GF2 (*'Gap-Filling for Grammar in French'*) is released as a research benchmark for the introduced task.



- We propose and train a suitable neural network architecture for the task, and show that conditioning the model’s output for a given input text on an example exercise of the envisioned exercise type, leads to an increased effectiveness, compared to an example-independent baseline model. Additionally we analyse the model’s ability to disentangle elementary exercise types, without being explicitly trained to do so, and we observe that it can recognize types to some extent, especially for the most commonly occurring types in the test set.

## 2 Gap-filling Exercise Creation as a Span Detection Task

This section describes the particular prediction task this paper focuses on. We cast the creation of a French gap-filling exercise from an input text as a *binary span detection task*: the goal is detecting each span (i.e., consecutive sequence of tokens) that represents a correct gap. For clarity, we left out creating the ‘hint’ (e.g., the infinitive for verbs) which would make it a finalized gap-filling exercise, as it is considered less challenging and may deviate attention from the core problem of identifying the correct spans.

Figure 1 shows two example gap-filling exercises, with indication of the ground truth spans in green (and with solid underline). We denote the distinguishing feature of each gap as its *gap type* (e.g., the tense *futur simple* for each of the valid tags in Example 1). An exercise typically covers multiple gap types, and the particular combination that characterizes a given exercise is called its *exercise type*. As such, many different exercise types can be constructed, and some may be unseen in the training data. For example, Example 2 (again in Fig. 1) combines three tenses (*imparfait*, *passé composé*, and *conditionnel présent*), which constitutes its exercise type. However, the same text could have been enriched with different gaps, corresponding to a different exercise type. In fact, our test set of one hundred exercises, for which we annotated gap types in terms of 12 elementary verb tenses, covers a total of 35 such composite exercise types.

Considering the lack of information regarding the exercise types for the training exercises, we further define the task we are examining more precisely. The objective is to detect the valid spans (i.e., spans that will be designated as gaps) of a given flat *input* text that mimics the same underly-

ing exercise type as an example gap-filling exercise, which we denote as the *exemplar*. This exemplar serves as an indirect reference for the model to understand the desired exercise type. By utilizing this approach, we can better inform the model about the desired exercise type while accounting for the lack of exercise information available.

Note that our goal is working with real-world data. Our training data contains gap-filling examples following particular unknown exercise types. Moreover, teachers appear to not always select every possible span that satisfies the exercise type. We saw cases in our dataset (cf. Section 4.1), where the same verb occurring twice in the same form would be selected as a valid gap only once. Such real-world ‘inconsistencies’ contribute to the challenging nature of learning from such data without additional annotations.

## 3 Example-aware span detection model

This section describes our baseline model and proposed example-aware gap detection model. Figure 2 provides a schematic overview. We first detail the part indicated as *Baseline model*, inside the smaller dashed box, followed by the part that encodes the exemplar, which leads to the full model.

**Baseline model:** An input text  $\mathbf{t}$ , consisting of  $N$  tokens  $\mathbf{t} = [t_0, t_1, \dots, t_{N-1}]$  is encoded by a transformer based masked language model (MLM), in our experiments the multilingual XLM-RoBERTa (Conneau et al., 2019). From the corresponding transformer outputs  $[\mathbf{h}_0, \mathbf{h}_1, \dots, \mathbf{h}_{N-1}]$  (with  $\mathbf{h}_i \in \mathbb{R}^k$ ,  $i=0 \dots N-1$ ), vector representations are constructed for all possible spans inside the input sequence, up to a certain length (in our experiments 12 tokens). The goal is then to make a binary prediction in terms of valid gaps, for each of these spans. In particular, for a span  $\varsigma = [t_{\text{start}}, \dots, t_{\text{end}}]$  with endpoint tokens  $t_{\text{start}}$ ,  $t_{\text{end}}$  and width  $|\varsigma| = (\text{end} - \text{start} + 1)$  in the input text, the corresponding span representation  $\mathbf{h}_{\varsigma}$  is constructed as

$$\mathbf{h}_{\varsigma} = \text{FFNN}(\mathbf{h}_{\text{start}} \oplus \mathbf{h}_{\text{end}} \oplus \mathbf{h}_{|\varsigma|})$$

in which  $\oplus$  represents vector concatenation,  $\mathbf{h}_{|\varsigma|}$  corresponds to a span width embedding, jointly learned with the model, and FFNN is a fully connected feed-forward model with a single hidden layer, ReLU activation, and output dimension  $k$ .

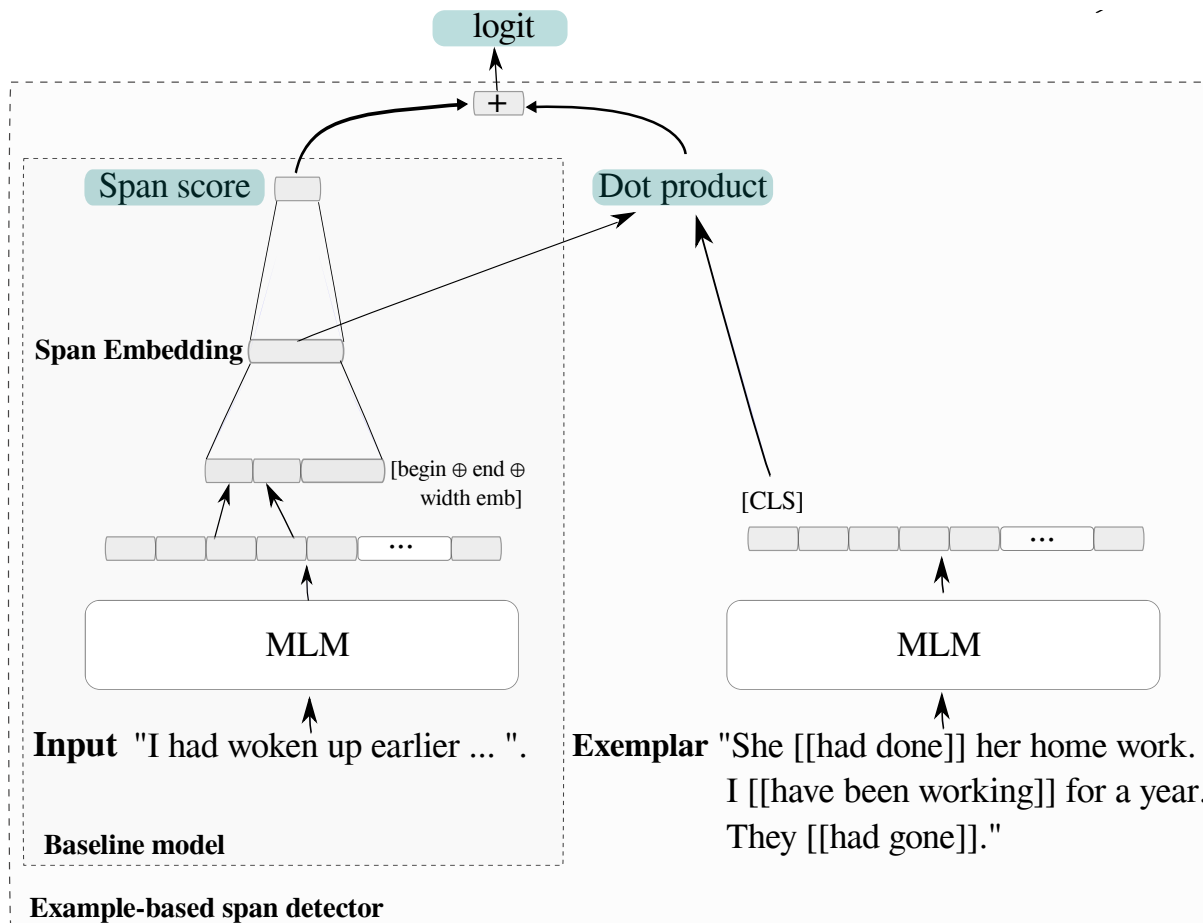


Figure 2: Example-aware gap detection model architecture.  $\oplus$  denotes concatenation. In general, the model considers all possible spans up to a maximum width, but we depict here only one span from the input for brevity.

The XLM-ROBERTa output representations  $\mathbf{h}_{\text{start}}$  and  $\mathbf{h}_{\text{end}}$  of the start and end token of  $\varsigma$  are concatenated with the span width embedding  $\mathbf{h}_{|\varsigma|}$ , and transformed through FFNN into the  $k$ -dimensional span representation  $\mathbf{h}_{\varsigma}$ . The probability of span  $\varsigma$  representing a valid gap is modeled as

$$p_{\text{base}}(\varsigma) = \sigma(\mathbf{w} \cdot \mathbf{h}_{\varsigma} + b)$$

in which the trainable parameters  $\mathbf{w}$  and  $b$  are a  $k$ -length coefficient vector and bias, respectively,  $\sigma$  is the sigmoid function, and  $\cdot$  represents the dot product. The baseline model is trained by minimizing the cross entropy loss between each span’s score  $p_{\text{base}}(\varsigma)$  and its label (1 for valid gaps, 0 otherwise). At inference, spans are predicted as gaps as soon as  $p_{\varsigma} \geq 0.5$ .

**Example-aware gap detection model:** As shown in Fig. 2, our example-aware model is a direct extension of the baseline model which by construction makes example-unaware predictions. The same MLM that encodes the input, is now

used to also encode the exemplar, which contains the example exercise text as well as the correct gap information. The latter is added by surrounding each gap with the special tokens ‘[’ and ‘]’ (as seen in the figure). Details on how the examples are chosen, are provided in Section 4.2. The exemplar representation  $\mathbf{h}_{\text{exemplar}}$  is obtained as the MLM’s [CLS] representation<sup>3</sup>.

We then quantify the compatibility of each span  $\varsigma$  in the input text with the exemplar, through the dot product  $\mathbf{h}_{\text{exemplar}} \cdot \mathbf{h}_{\varsigma}$  of their respective representations. In a direct extension of the baseline model, it leads to the proposed model for the probability  $p_{\text{example-aware}}(\varsigma)$  that  $\varsigma$  represents a valid gap:

$$p_{\text{example-aware}}(\varsigma) = \sigma(\mathbf{h}_{\varsigma} \cdot \mathbf{w} + \mathbf{h}_{\varsigma} \cdot \mathbf{h}_{\text{exemplar}} + b)$$

<sup>3</sup>[CLS] is a special token that is prepended to the input, and its corresponding output representation is pretrained to represent the entire sequence that is used for classification tasks

## 4 Empirical validation on real-world data

In this section, we first introduce the dataset that we will publicly release. Then, we explain how we train our models and use them for inference. Finally, we describe the strategies we adopted to evaluate the effectiveness of our models.

### 4.1 GF2 dataset: Gap-Fill for Grammar in French

We denote our new dataset as “Gap-Filling for Grammar in French” (GF2). It was contributed by Televic Education<sup>4</sup>, and gathered through its education platform assessmentQ<sup>5</sup>. AssessmentQ is a comprehensive online platform for interactive workforce learning and high-stakes exams. It allows teachers to compose their questions and answers for practice and assessment. As a result, the dataset is made up of a real-world set of gap-filling grammar exercise questions for French, manually created by experts. We cleaned and preprocessed the data before we could use it to train our models. First, organizational metadata information was removed. Other elements that we removed are the hints within the body of the text that could easily give away the gap positions, as well as inline instructions (if present) about the exercise type. Second, we automatically stripped off HTML tags from the documents. Our final dataset contains a total of 768 exercise documents, in which a total of 5,530 spans are tagged as gaps. The exercises were randomly split into 618 train documents, and 50 and 100 for validation and test, respectively. Table 1 summarizes GF2’s descriptive statistics.

For the validation and test exercises, we made an extra manual effort to enrich each of the existing gaps with their gap type. Our annotations reflect the fact that the data contains a mix of verb and non-verb gaps. Every gap has an associated word type attribute (e.g. adverb, adjective, verb) and in case of verbs a tense attribute. In what follows we zoom in on the verb gaps and consider the tense as the main gap type. The bottom half of Table 1 shows the frequency of occurrence for the main verb types in the development and test documents. We use these annotations to get insights into the dataset and to evaluate the properties of our models (see Section 5). Note that the examples shown in Fig. 1 are actual entries from the GF2 dataset.

<sup>4</sup><https://www.televic.com/en/education>

<sup>5</sup><https://www.televic-education.com/en/assessmentq>

### 4.2 Training and inference

Our baseline model is relatively straightforward to train. We designate all spans indicated as gaps in our training data as valid gaps, which are considered positive examples. Conversely, any spans that are not indicated as gaps are labeled as negatives. We train our model by minimizing the cross entropy loss between each span’s predicted score and its label as described in Section 3. However, training our example-aware model poses a challenge due to the lack of knowledge regarding the exercise types of the training exercises. Using one exercise as an example and another exercise of the same type as the input, along with the corresponding targets, is not therefore feasible. Instead, we make the assumption that exercises are generated by teachers who consistently follow the underlying exercise type throughout the entire exercise. As a result, we divide the training exercises into two parts: one part is used as an exemplar, and the other part serves as the actual input, for which the gaps are assumed to follow the same exercise type.

To this end, we first segment each document in the training set into a list of sentences, along with their corresponding target gap positions. We create a new (exemplar, input) training pair by sampling one sentence to be used as the input, and uniformly sampling one up to  $m$  sentences from the remaining sentences within the same document to be used as the exemplar. The exemplar is constructed by concatenating these sampled sentences, with the addition of special symbols denoting the gap locations. (See Appendix A for details.) These are the positive training examples that encourage the model to correctly learn predicting example-aware gaps. However, to facilitate efficient learning, it is crucial to also provide negative examples on which the model should not predict gaps. To create such negative training instances, a sentence is sampled as input from the considered document, but its span targets are set to zero (no gaps), and the negative exemplar is composed as before (including indicating the gaps), but by sampling sentences from a randomly selected *other* training exercise. There is risk of incidentally creating false negative training examples, if the exemplar gaps correspond with left-out gaps in the input. However, negative exemplars appeared important for obtaining a suitable model.

We determine the optimal proportion of negative to positive instances for training our models by em-

Table 1: Statistics of the FG2 dataset and breakdown into key verb tenses (gap types) in the validation and test split. For the train split we only know gap spans, not their types, since they are not labelled.

	<b>Train</b>	<b>Dev</b>	<b>Test</b>
# Documents	618	50	100
# Sentences	4786	378	707
# Gaps	4518	365	647
Subjonctif Présent (SPR)	UNK	1	28
Passé Composé (participe passé) (PCP)	UNK	31	8
Passé Composé (PC)	UNK	84	108
Imparfait (IM)	UNK	8	46
Conditionnel Présent (CPR)	UNK	23	92
Passé Récent (PR)	UNK	0	12
Futur Proche (FP)	UNK	1	9
Futur Simple (FS)	UNK	8	49
Indicatif Présent (IP)	UNK	126	144
Conditionnel Passé (CPA)	UNK	0	3
Impératif (IMP)	UNK	12	26
Plus-que-parfait (PQ)	UNK	0	1

ploying a fine-tuning approach utilizing the macro F1 score as the evaluation metric on the validation set. This increases the impact of the rarer gap types in the metric, and therefore in the final model, which we considered important for practical use. Other choices could have been made, however. Ultimately, the final model is trained on the union of the training and validation splits, using the optimal proportion determined via the fine-tuning process.

During inference, we use our trained model to predict the gap positions for an input text that is implicitly conditioned on the target exercise type through the exemplar.

**Implementation and training details:** We implement our models using pytorch and Huggingface. We initialize our MLM encoders with xlm-roberta-base. To avoid extensive hyperparameter tuning, we made the following choices; a learning rate of 2e-5 in combination with the robust Adam optimizer. We use a batch size of 16 and train our models for 30 epochs. We consider all spans up to a maximum length 12 and we set  $k$ , the number of sentences per exemplar to 3.

### 4.3 Evaluation setup

In order to assess and analyze the performance of the baseline and the example-aware model, we design two evaluation strategies that look at different effectiveness aspects.

**Binary gap prediction evaluation:** the primary objective of our model is to mimic the real-world setting where gap labels are not given. We measure how well our models predict gap positions (i.e., gap or no-gap decisions for all input spans). To do this, we split up each of the exercise documents in our test into two parts that are roughly the same size, given that by assumption they then represent the same exercise type. We calculate the automated metrics by using one half as the exemplar and the second as the input text to our model. We repeat this process by exchanging the roles of the parts. It is worth noting that we excluded one-sentence test documents (i.e., because they can not be chunked into two parts), which amount to 16% of the total test documents. However, since most of the excluded sentences (i.e., one-line documents) only had one gap, we only removed 2.7% of the total gaps in the test set.

**Gap type disentangling evaluation:** The goal of the second evaluation setting is to analyze how well the model has learned to disentangle individual gap types, despite not being explicitly trained to do so. This analysis is based on the assumption that a model that scores high on that aspect, would be stronger in dealing with new or rare exercise types. Potentially even at creating new combinations of existing exercises. This is an aspect we plan to study further when designing more advanced models in future research. To this end, we construct

Table 2: Tense disentangling ability in terms of precision, recall, and F1 (in %) on the test set, as reported for each key verb tenses (with on the right their support, i.e., number of occurrences). We also show the macro F1 score for the static baseline (*baseline*) and our proposed example-aware gap prediction (*ours*).

Tenses	<i>Baseline</i>			<i>Ours</i>			Support
	P	R	F1	P	R	F1	
SPR	5.0 $\pm$ 0.3	78.6 $\pm$ 8.9	9.4 $\pm$ 0.6	7.5 $\pm$ 0.2	81.0 $\pm$ 12.5	13.7 $\pm$ 0.4	28
PCP	0.1 $\pm$ 0.1	4.2 $\pm$ 6.3	0.2 $\pm$ 0.3	12.6 $\pm$ 4.1	62.5 $\pm$ 12.5	20.7 $\pm$ 6.2	8
PC	21.3 $\pm$ 1.2	86.4 $\pm$ 3.7	34.2 $\pm$ 1.8	64 $\pm$ 9.4	86.1 $\pm$ 1.9	73.1 $\pm$ 5.5	108
IM	9.3 $\pm$ 0.4	88.4 $\pm$ 3.7	16.2 $\pm$ 0.8	12.0 $\pm$ 2.5	78.3 $\pm$ 10.9	20.9 $\pm$ 3.9	46
CPR	19.9 $\pm$ 0.5	94.5 $\pm$ 2.9	32.8 $\pm$ 0.8	28.3 $\pm$ 2.9	92.4 $\pm$ 4.7	43.2 $\pm$ 3.1	92
PR	2.7 $\pm$ 0.1	100.0 $\pm$ 0.0	5.3 $\pm$ 0.1	9.7 $\pm$ 2.0	100.0 $\pm$ 0.0	17.7 $\pm$ 3.3	12
FP	1.6 $\pm$ 0.0	77.7 $\pm$ 0.0	3.1 $\pm$ 0.1	6.0 $\pm$ 0.9	77.8 $\pm$ 0.0	11.1 $\pm$ 1.5	9
FS	9.9 $\pm$ 0.3	88.5 $\pm$ 1.7	17.8 $\pm$ 0.5	13.6 $\pm$ 1.1	84.4 $\pm$ 10	23.3 $\pm$ 1.7	49
IP	24.6 $\pm$ 1.2	75.0 $\pm$ 4.3	37.1 $\pm$ 1.9	32.0 $\pm$ 1.4	66.2 $\pm$ 11.9	42.9 $\pm$ 2.4	144
CPA	0.1 $\pm$ 0.1	11.1 $\pm$ 16	0.2 $\pm$ 0.3	0	0	0	3
IMP	5.2 $\pm$ 0.3	88.5 $\pm$ 2.2	9.9 $\pm$ 0.5	16.8 $\pm$ 1.7	84.6 $\pm$ 3.9	25.3 $\pm$ 2.1	26
PQ	0.2 $\pm$ 0.0	100.0 $\pm$ 0.0	0.5 $\pm$ 0.0	0.6 $\pm$ 0.1	100 $\pm$ 0.0	1.2 $\pm$ 0.2	1
<b>Macro F1</b>		<b>13.9</b>			<b>24.4</b>		

a small set of 12 exemplars, one for each of the key verb tenses, by randomly selecting them from the original data and subsequently removing them from the train/validation/test splits. Each exemplar comprises multiple sentences, all of which are homogeneously annotated with the same intended verb type, which will serve as the desired homogeneous exercise type. We evaluate our model on every sentence of the test set, by prompting it with each of these 12 fixed exemplars. Based on the gap types we annotated on the test set, we can then compute the precision, recall and F1 score for each of these 12 tenses.

## 5 Experimental Results

In this section, we provide evidence of the effectiveness of our proposed model by reporting and discussing the experimental results. Table 3 summarizes the binary gap prediction evaluation of the baseline vs. the example-aware model on the test set. We report our results as the mean and standard deviation over five runs, each using a different random seed for model training. The proposed example-aware model (denoted as *ours*) consistently outperforms the example-unaware *baseline* on all metrics. In general, there is an absolute gain of 8 percentage points in F1 for the proposed model in comparison with the baseline, achieving an average F1 score of 82.4%. This confirms our intention when designing the model, that providing exam-

ple exercises leads to an increased effectiveness in terms of predicting gap positions compared to the static baseline model.

Table 3: Overall binary gap prediction in terms of precision, recall, and F1 (in %) on the test set. Results shown for the static baseline (*baseline*) and our proposed example-aware gap prediction (*ours*).

	<b>Precision</b>	<b>Recall</b>	<b>F1</b>
<i>Baseline</i>	74.87 $\pm$ 2.44	73.11 $\pm$ 2.00	73.92 $\pm$ 0.49
<i>Ours</i>	84.30 $\pm$ 1.70	80.74 $\pm$ 1.80	82.40 $\pm$ 0.20

In Table 2, we show the evaluation of our models in their ability to disentangle the 12 main verb types. We observe that for the tenses with relatively higher support, the example-aware model outperforms the baseline with certainty as demonstrated by the individual F1 scores.

The overall macro F1 score for the example-aware model stands at 24.4%, which is low in absolute value, but considerably higher than the baseline’s macro F1 score of 13.9%. We observe that the proposed model is able to recognize verb types such as passé composé (PC), imparfait (IM), and conditionnel présent (CPR) to some extent with F1 scores of 73%, 43%, and 42%, respectively. However, the low overall scores are not unexpected, because the models are not trained to recognize gap types. Furthermore, some tenses are either

very rare (e.g., PQ, CPA, PCP) as indicated by their support, or may appear mainly in combination with other exercise types. This makes achieving a better resolution in disentangling gap types without any explicit gap labels during training an inherently difficult task.

## 6 Conclusion

In this paper, we introduced a new task within the general challenge of training models to automatically create new exercises for use in education, based on existing exercises and without requiring additional manual annotations.

In particular, we introduced a dataset and associated prediction task, focusing on detecting gaps within a given input text, without knowledge of the exact exercise type, by only relying on an example exercise. We proposed an example-aware neural network model designed for this task, and compared it with a baseline model that does not take into account any example of the desired exercise type. We found that our example-aware model outperforms the baseline model not only in predicting gaps, but also in disentangling gap types despite not being explicitly trained on that task. Our real-world GF2 dataset of French gap-filling exercises will be publicly released together with the code to reproduce the presented empirical results.

The presented work fits with our pursuit towards supporting personalized learning experiences by either suggesting existing or generating new exercises that are tailored to students' needs. Teachers could also benefit from an increased efficiency in creating new exercises. For example, they could make many and diverse drill and practice exercises on chunks of text based on existing standard exercise types without having to provide extra meta-data information such as instructions. We hope our benchmark dataset and task will spark new research in the CL and Educational NLP community.

## Limitations

We identify two limitations of the current work and make suggestions for future directions. First, while our proposed method is language-agnostic in principle, our evaluation is limited to our French benchmark dataset. Expanding our approach to encompass other languages would bring new and interesting challenges for further investigation. Second, despite topic diversity within our exercise documents (e.g., the first example in Fig. 1 consists of

independent sentences, while the second is a coherent text centered around the same topic.), it would be interesting to quantify the degree of topical bias introduced during our training process and its impact on our binary task evaluation. For future work, we first aim to adapt seq2seq models for our task particularly text-to-text models such as T5 (Raffel et al., 2020). There is also potential to explore different prompting strategies for large language models (LLMs), when generating gap-filling grammar exercises. For instance, the utilization of chain-of-thought prompting (Wei et al., 2022), which involves generating intermediate steps before producing the final response, could be explored for generating grammar exercises. Additionally, an interesting future study would involve investigating the number of example demonstrations that LLMs require in order to accurately mimic example gap exercises.

## Ethics Statement

In this research, we posit that the dataset and models introduced are of low-risk in terms of potential harm to individuals. The dataset used is a curated selection of existing educational content enriched with meta-data, and we are confident that our compilation of the dataset has not introduced any additional ethical risks. However, it is crucial to emphasize the need for accountability and the establishment of clear guidelines for the deployment of grammar generation models, such as the ones benchmarked in this paper, for educational purposes.

It should be noted that our models are derived from general-purpose neural language encoders that have been trained on real-world data, which may contain biases or discriminatory content (Bommasani et al., 2021). As a result, our models may have inherited some of these biases and could potentially base their prediction on such biased information. Therefore, it is imperative for educators and researchers to thoroughly consider these ethical issues and ensure that the generated grammar questions align with educational goals and do not perpetuate harmful biases.

Educators should retain the final authority in accepting or modifying grammar question suggestions generated by such models, keeping their educational goals in mind (e.g., in terms of formative and especially summative assessment). In practice, these models are designed to enhance teachers' effi-

ciency in preparing teaching materials, rather than replacing teachers in any way. An important benefit of using AI-supported question generation with increased efficiency is the potential for personalized approaches towards students.

## Acknowledgements

This work was funded by VLAIO (‘Flanders Innovation & Entrepreneurship’) in Flanders, Belgium, through the *imec-icon* project AIDA (‘AI-Driven e-Assessment’). This research also received funding from the Flemish Government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” programme. We would like to thank the AIDA partners Televic Education and WeZooz Academy for contributing data and use cases.

## References

- Naveed Afzal and Ruslan Mitkov. 2014. Automatic generation of multiple choice questions using dependency-based semantic relations. *Soft Computing*, 18(7):1269–1281.
- Manish Agarwal and Prashanth Mannem. 2011. Automatic gap-fill question generation from text books. In *Proceedings of the sixth workshop on innovative use of NLP for building educational applications*, pages 56–64.
- Maha Al-Yahya. 2011. Ontoque: a question generation engine for educational assesment based on domain ontologies. In *2011 IEEE 11th International Conference on Advanced Learning Technologies*, pages 393–395. IEEE.
- Semere Kiros Bitew, Amir Hadifar, Lucas Sterckx, Johannes Deleu, Chris Develder, and Thomas De-meester. 2022. Learning to reuse distractors to support multiple choice question generation in education. *IEEE Transactions on Learning Technologies*.
- Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosse-lut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. [Electra: Pre-training text encoders as discriminators rather than generators](#). In *International Conference on Learning Representations*.
- Ricardo Conejo, Eduardo Guzmán, and Monica Trella. 2016. The siette automatic assessment environment. *International Journal of Artificial Intelligence in Education*, 26(1):270–292.
- 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*.
- Thanasis Daradoumis, Joan Manuel Marquès Puig, Marta Arguedas, and Laura Calvet Liñan. 2019. Analyzing students’ perceptions to improve the design of an automated assessment tool in online distributed programming. *Computers & Education*, 128:159–170.
- Barbara Gross Davis. 2009. *Tools for teaching*. John Wiley & Sons.
- 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*.
- Xinya Du, Junru Shao, and Claire Cardie. 2017. [Learning to ask: Neural question generation for reading comprehension](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352, Vancouver, Canada. Association for Computational Linguistics.
- Mariano Felice, Shiva Taslimipour, and Paula Buttery. 2022. [Constructing open cloze tests using generation and discrimination capabilities of transformers](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1263–1273, Dublin, Ireland. Association for Computational Linguistics.
- Takuya Goto, Tomoko Kojiri, Toyohide Watanabe, Tomoharu Iwata, and Takeshi Yamada. 2010. Automatic generation system of multiple-choice cloze questions and its evaluation. *Knowledge Management & E-Learning: An International Journal*, 2(3):210–224.
- Jennifer Hill and Rahul Simha. 2016. Automatic generation of context-based fill-in-the-blank exercises using co-occurrence likelihoods and google n-grams. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 23–30.
- John Lee and Stephanie Seneff. 2007. Automatic generation of cloze items for prepositions. In *Eighth Annual Conference of the International Speech Communication Association*.
- Anna Malinova and Olga Rahneva. 2016. Automatic generation of english language test questions using mathematica. In *CBU International Conference Proceedings*, volume 4, pages 906–909.

- Edison Marrese-Taylor, Ai Nakajima, Yutaka Matsuo, and Ono Yuichi. 2018. Learning to automatically generate fill-in-the-blank quizzes. *arXiv preprint arXiv:1806.04524*.
- 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.
- John W Oller Jr. 1973. Cloze tests of second language proficiency and what they measure 1. *Language learning*, 23(1):105–118.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.
- Andreas Papasalouros, Konstantinos Kanaris, and Konstantinos Kotis. 2008. Automatic generation of multiple choice questions from domain ontologies. *e-Learning*, 1:427–434.
- Laura Perez-Beltrachini, Claire Gardent, and German Kruszewski. 2012. Generating grammar exercises. In *The 7th Workshop on Innovative Use of NLP for Building Educational Applications, NAACL-HLT Workshop 2012*, pages 147–157.
- Juan Pino, Michael Heilman, and Maxine Eskenazi. 2008. A selection strategy to improve cloze question quality. In *Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains. 9th International Conference on Intelligent Tutoring Systems, Montreal, Canada*, pages 22–32.
- Debra Pugh, Andre De Champlain, Mark Gierl, Hollis Lai, and Claire Touchie. 2016. Using cognitive models to develop quality multiple-choice questions. *Medical teacher*, 38(8):838–843.
- 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. *The Journal of Machine Learning Research*, 21(1):5485–5551.
- Vanja Slavuj, L Nacinovic Prskalo, and M Brkic Bakaric. 2021. Automatic generation of language exercises based on a universal methodology: An analysis of possibilities. *Bulletin of the Transilvania University of Brasov. Series IV: Philology and Cultural Studies*, pages 29–48.
- Katherine Stasaski and Marti A Hearst. 2017. Multiple choice question generation utilizing an ontology. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 303–312.
- Eiichiro Sumita, Fumiaki Sugaya, and Seiichi Yamamoto. 2005. Measuring non-native speakers’ proficiency of english by using a test with automatically-generated fill-in-the-blank questions. In *Proceedings of the second workshop on Building Educational Applications Using NLP*, pages 61–68.
- Bo Sun, Yunzong Zhu, Yongkang Xiao, Rong Xiao, and Yungang Wei. 2018. Automatic question tagging with deep neural networks. *IEEE Transactions on Learning Technologies*, 12(1):29–43.
- Yuni Susanti, Takenobu Tokunaga, Hitoshi Nishikawa, and Hiroyuki Obari. 2017. Evaluation of automatically generated english vocabulary questions. *Research and practice in technology enhanced learning*, 12(1):1–21.
- Wilson L Taylor. 1953. “cloze procedure”: A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.
- Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. [Chain of thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems*.

## A Training details

In this section we detail our training procedure. As depicted in Fig. 3, we first split training exercises into list of sentences, along with their corresponding gap position indications. In order to create new (input, exemplar) pair, we sample 1 sentence from the sentence list to be used as our *input* text, and we uniformly sample 1 up to  $m$  (we set  $m = 3$ ) sentences from the remaining sentence list to be used as our exemplar. We form our exemplar by concatenating all the sampled sentences with gap positions indicated by special tokens “[[” and “]]”. Then our model is trained by minimizing the binary cross entropy (BCE) loss between predicted gaps and their target labels (1 for valid gaps, and 0 otherwise).



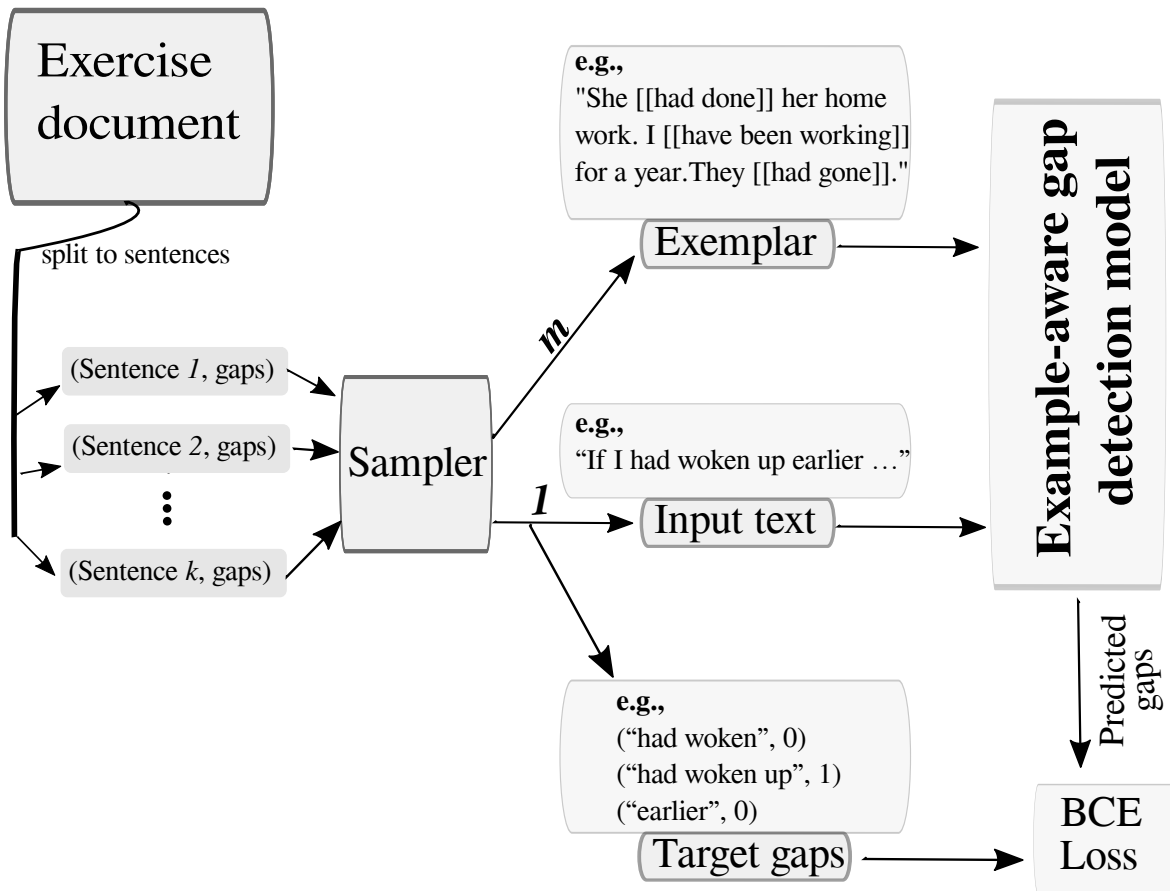


Figure 3: Training procedure of our example-aware gap detection model. First, we split exercise documents into list of sentences. Then we create (input, exemplar) training pairs that will be used by our model. We use one sentence as an input, while the exemplar is made up of sentences that are uniformly sampled from the remaining sentences. The exemplar is constructed by concatenating the  $m$  sampled sentences. The special symbols “[[” and “]]” in the exemplar indicate the gap positions. Binary cross entropy (BCE) loss is used to train our models.

# Evaluating Reading Comprehension Exercises Generated by LLMs: A Showcase of ChatGPT in Education Applications

Changrong Xiao<sup>1</sup>, Sean Xin Xu<sup>1</sup>, Kunpeng Zhang<sup>2</sup>, Yufang Wang<sup>3</sup>, Lei Xia<sup>4</sup>

<sup>1</sup>School of Economics and Management, Tsinghua University

<sup>2</sup>Department of Decision, Operations & Information Technologies, University of Maryland

<sup>3</sup>Beijing Xicheng Educational Research Institute

<sup>4</sup>Shawn Tech

xcr21@mails.tsinghua.edu.cn, xuxin@sem.tsinghua.edu.cn,

kpzhang@umd.edu, wangwang7587@163.com, xialei@shawntech.com.cn

## Abstract

The recent advancement of pre-trained Large Language Models (LLMs), such as OpenAI's ChatGPT, has led to transformative changes across fields. For example, developing intelligent systems in the educational sector that leverage the linguistic capabilities of LLMs demonstrates a visible potential. Though researchers have recently explored how ChatGPT could possibly assist in student learning, few studies have applied these techniques to real-world classroom settings involving teachers and students. In this study, we implement a reading comprehension exercise generation system that provides high-quality and personalized reading materials for middle school English learners in China. Extensive evaluations of the generated reading passages and corresponding exercise questions, conducted both automatically and manually, demonstrate that the system-generated materials are suitable for students and even surpass the quality of existing human-written ones. By incorporating first-hand feedback and suggestions from experienced educators, this study serves as a meaningful pioneering application of ChatGPT, shedding light on the future design and implementation of LLM-based systems in the educational context.

## 1 Introduction

Reading comprehension is a vital skill that English learners need to develop and master. Chinese middle school students, for instance, are required to do numerous English practices, including reading at least 150,000 words of supplemental materials to enhance their reading abilities, as mandated by the English Curriculum Standards.

Through interviews with experienced English teachers in Beijing, we discovered a challenge faced by both educators and students: the repeated use of outdated reading materials, with only minor modifications made, if any. For instance, Grade 8 students are likely to practice the same exercises

used by their predecessors in the previous academic year (currently Grade 9 students). English teachers believe that offering up-to-date, engaging reading exercises tailored to each student's capabilities and interests can spark their enthusiasm for learning and ultimately boost their English proficiency.

However, obtaining a large collection of diverse, customized, high-quality English reading exercises proves to be a non-trivial task. There is an abundance of articles in newspapers, magazines, textbooks, and children's books from English-speaking countries that could serve as potential sources of reading materials for middle school students. Nonetheless, adjustments and rewrites are typically necessary due to variations in topic, length, and difficulty level. Moreover, even for veteran teachers, crafting appropriate exercise questions based on textual materials is still not easy.

Pre-trained Large Language Models (LLMs) have been proposed by researchers as a means to address this labor-intensive and unscalable issue (Zhai, 2022; Dwivedi et al., 2023). Reading comprehension exercises typically consist of two components: a lengthy, coherent passage and several multiple-choice questions that align with its content. To generate such exercises, it is essential for LLMs to possess an advanced understanding and inference ability of human language. While the generation of long texts (such as stories, news articles, and poems) (Li et al., 2021) and question-and-answer (Q&A) pairs (Kurdi et al., 2020) have been extensively studied, existing task-specific models fall short of meeting our needs. For instance, the generated content still remains distinguishable from human-written text, and the level of personalization for different learners is inadequate (Kurdi et al., 2020), making these models unsuitable for direct application in educational settings.

Recently, OpenAI released ChatGPT<sup>1</sup>, a versatile and interactive chatbot that outperforms state-

<sup>1</sup><https://openai.com/blog/chatgpt>

of-the-art models in various NLP tasks, even in zero-shot or few-shot scenarios. This powerful LLM presents numerous opportunities for education, including the creation of reading materials and customized practice questions. In this study, we attempt to develop a system for middle school teachers and students that leverages ChatGPT to generate reading comprehension exercises. Guided by carefully crafted prompts, ChatGPT can produce personalized reading passages and multiple-choice questions of high quality. To assess the generated exercises and the overall system, human evaluators (comprising students, teachers, and native speakers) conducted an extensive analysis, determining that the system holds promise for implementation in middle schools and has the potential to make a significant educational impact. In summary, this study makes threefold contributions:

- We fully leverage the capabilities of the state-of-the-art LLMs to tackle complex and compound tasks, integrating them within a carefully designed education system<sup>2</sup>. The reading passages and exercise questions generated by our system significantly surpass the quality of those produced by previous models, with some even exceeding the standard of human-written textbook exercises.
- To the best of our knowledge, our reading exercise generation system is among the first applications of ChatGPT in the education context. The system has been utilized by middle school English teachers, making real impacts in schools.
- We gather feedback from both experts and general users regarding the efficacy of our system. We believe this is valuable, as there are few instances of ChatGPT applications being employed in real-world educational settings. Our findings offer insights for future researchers and practitioners to develop more effective AI-driven educational systems.

## 2 Related Work

**LLM and Controllable Text Generation** With the emergence of Transformers (Vaswani et al., 2017), LLMs have been performing remarkably well and showing considerable progress across a

<sup>2</sup>The codes for our system is available at <https://github.com/Xiaochr/Reading-Exercise-Generation-System>.

variety of NLP tasks (Qiu et al., 2020). For example, OpenAI’s GPT series models are powerful LLMs that perform well in long open-ended text generation. While they are able to generate texts of high fluency, researchers have found that as the generated text gets longer, it starts to wander, switch to unrelated topics, and become incoherent (Rashkin et al., 2020). By fine-tuning with specific domain data or applying some plug-and-play approaches like PPLM (Dathathri et al., 2020), LLMs will obtain some controllability and generate more coherent text, though the quality is still limited.

ChatGPT is developed on the foundation of GPT-3.5 or GPT-4 architectures, with the inclusion of additional human-directed instructions for enhanced performance. It possesses robust in-context learning capabilities, enabling it to interpret requirements specified in input prompts without the need for additional information (zero-shot learning), or by utilizing a minimal number of provided examples (few-shot learning). Even without massive domain knowledge, ChatGPT is able to follow human instructions and generate text of higher quality. For instance, to generate a 200-word reading passage on the topic of school life, one simply needs to specify the subject and length requirements in the prompt to ChatGPT.

**ChatGPT in Education** With the thriving of AI technology, its applications in education have been increasing, transforming ways of teaching and learning (Zhang and Aslan, 2021). Recognizing the surprising capacity of LLMs, such as ChatGPT, researchers have been discussing their enormous potential impacts in various educational scenarios (Zhai, 2022). Some studies (Dwivedi et al., 2023; Pettinato Oltz, 2023) suggested that ChatGPT can provide students with basic educational materials. LLMs are trained on vast corpora created by humans to “learn” the language, and now they can “teach” human learners what they have already learned. Moreover, inherent to its chatbot characteristics, ChatGPT can function as a personal tutor, providing real-time feedback (Zentner, 2022), personalized evaluations and suggestions (Baidoo-Anu and Owusu Ansah, 2023; Zhang, 2023), and other learning supports (Dwivedi et al., 2023), such as improving the engagement and autonomy of students (Firat, 2023) and addressing the low teacher-student ratio problem (Chen et al., 2023).

On the other hand, the misuse of ChatGPT has existed since its release (Zhang et al., 2023). A poll

<sup>3</sup> done by Study.com (an online course provider) reveals that 89% of the participating students utilized ChatGPT for homework and 48% of them confessed to using ChatGPT for at-home tests. It is important and still under exploration to design suitable learning tasks and systems that can guide students to use ChatGPT properly as a helpful learning assistant.

**Evaluation of Long Text Generation** To evaluate the quality of the generated long text, researchers have developed several metrics, including Self-BLEU (Zhu et al., 2018) and  $n$ -gram repetition score (Welleck et al., 2020). They are often unreliable and inconsistent with human judgment (Belz et al., 2020). Therefore, human evaluation remains the gold standard for most long text generation tasks, even if it is expensive and time-consuming (Celikyilmaz et al., 2020).

Belz and Reiter (2006) grouped the common human evaluation approaches into intrinsic and extrinsic ones. Most current text generation tasks are measured with intrinsic human evaluations, where participants are asked to rate the quality of the generated text, either overall or along with some designed dimensions (e.g., fluency, coherence, and correctness) (Celikyilmaz et al., 2020). Likert and sliding scale are commonly used scoring methods, despite the many limitations (e.g., inconsistency, not straightforward) (Celikyilmaz et al., 2020). To address this, comparative approaches, such as ranking, have been proposed and found to achieve high inter-annotator agreement (Callison-Burch et al., 2007). On the other hand, the extrinsic evaluation measures how successful a system is in downstream tasks, from both a user’s success in a task and the system’s success in fulfilling its purpose (Celikyilmaz et al., 2020; Hastie and Belz, 2014).

### 3 Methods

#### 3.1 Reading passage Generation Baseline

We use a fine-tuned GPT-2 (Radford et al., 2019) with PPLM (Dathathri et al., 2020) control as the baseline method to generate reading passages. The two-stage development of the baseline model is shown in Figure 1.

In the first step, we fine-tune our base LLM, GPT-2 medium, using two reading datasets obtained from middle school teachers: supplemental

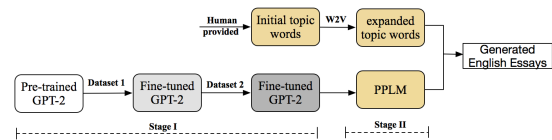


Figure 1: The fine-tuned GPT-2 + PPLM baseline

reading materials (Dataset 1) and textbook exercise passages that are currently used in middle schools (Dataset 2). We adopt a two-step fine-tuning strategy with varying learning rates to accommodate the distinct characteristics of each dataset. In the second step, we employ PPLM, a plug-and-play controllable text generation approach, to guide the fine-tuned language model in generating more coherent passages based on specified topic keywords. For more details, please refer to the Appendix A.

#### 3.2 ChatGPT for Reading Exercise Generation

Utilizing the impressive capabilities of ChatGPT, we manually design input prompts to generate high-quality reading comprehension passages without the need for fine-tuning or additional control methods. In this study, we produce textual content in two settings: zero-shot and one-shot, which allow us to control the output to varied degrees.

In the zero-shot setting, we instructed ChatGPT to be a helpful learning assistant capable of generating high-quality reading passages in the system prompt. We provided customized requirements within the conversation prompt, including length, genre, difficulty level, and topics. In addition to creating reading passages from scratch, teachers often source content from the web or other materials and seek to adapt them into suitable reading passages for students. Thus we added an extra requirement, a referenced passage, in the one-shot setting.

We also generate questions and corresponding answers for given passages using appropriate prompts. We set the number of questions, the number of options per question, and the question type for customization in the input prompt. ChatGPT can generate exercise questions based on either a passage it previously created or a passage provided by users. Moreover, an extra toxicity check is applied before the generated exercises are made available to teachers and students.

We will describe the process of reading exercise generation using ChatGPT and the design of appropriate prompts in Appendix B.

<sup>3</sup><https://futurism.com/the-byte/students-admit-chatgpt-homework>

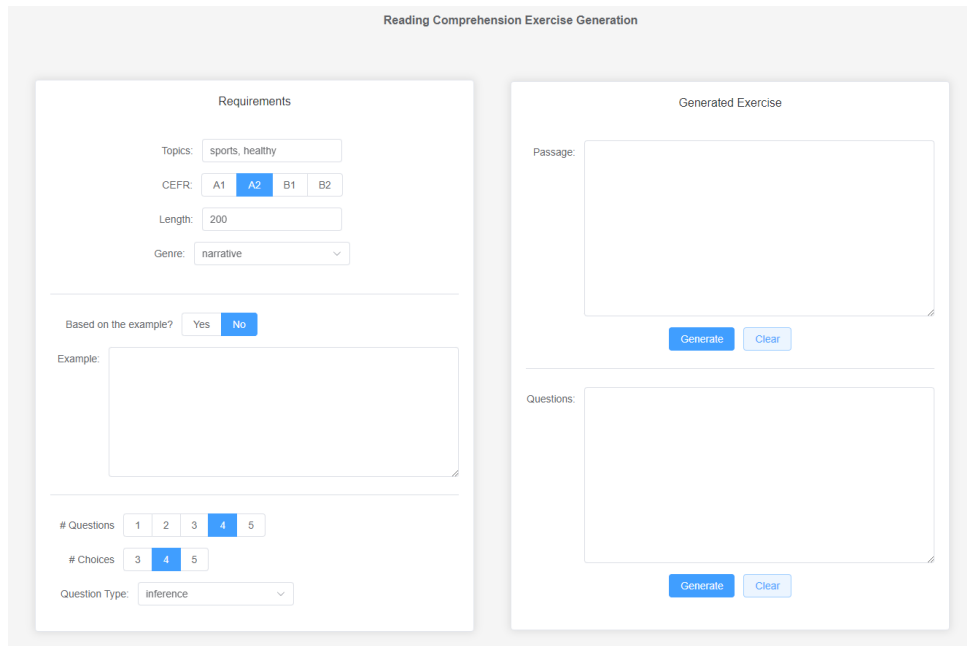


Figure 2: The screenshot of the system interface.

### 3.3 System Design

Catering to non-technical users such as middle school teachers and students, we integrate the features discussed in previous sections into a unified system with a graphical user interface. The prompts and API calls are managed at the system backend, while a user-friendly and straightforward interface (Figure 2) is designed for ease of use<sup>4</sup>.

On the left side of the interface, users can easily set their requirements, with each previously mentioned feature incorporated. The output reading passages and exercise questions are displayed on the right. These text areas are editable, allowing teachers to further modify the generated content to create a final version of exercises suitable for student practice.

## 4 Evaluation

In this section, we conduct extensive evaluations of our reading exercise generation system, which are visually depicted in Figure 3.

For reading passage quality evaluation, we randomly select 30 human-written reading passages from Dataset 2 (the reading exercises from textbooks), which are paired with an additional 60 passages: 30 produced by ChatGPT and 30 by the baseline model. This mixture of passages is shuffled and compiled into what we refer to as the

<sup>4</sup>To try the system demo online, please refer to our GitHub repository. We will keep the link to the demo up-to-date.

Reading Passages Example Set 1. We utilize both automatic evaluation metrics and human assessments (Section 4.1) in order to comprehensively evaluate these passages.

To further verify the high quality of ChatGPT passages, a series of one-to-one comparisons is conducted between passages produced by language models and their human-written counterparts. We select 10 human-written reading comprehension passages, distinct from the passages in the Reading Passages Example Set 1, and summarize the topic of each one. We then use these topics as guiding constraints to direct conditional text generation with both the GPT-2 + PPLM baseline and ChatGPT (zero-shot), resulting in passages mirroring the topics of the original human-written examples. Additionally, a one-shot variant of ChatGPT, using the human-written passage as a reference, is utilized to generate a third group of passages. To sum up, the Reading Passages Example Set 2 encompasses 10 original human-written passages, augmented with 30 generated passages that align with the same topics.

Moving to the evaluation of exercise question quality, we select 10 exercises containing reading passages and their associated questions from the textbook to serve as benchmarks. A new set of multiple-choice questions is generated based on the human-written passages using our system. Thus, these 10 reading passages and their corresponding

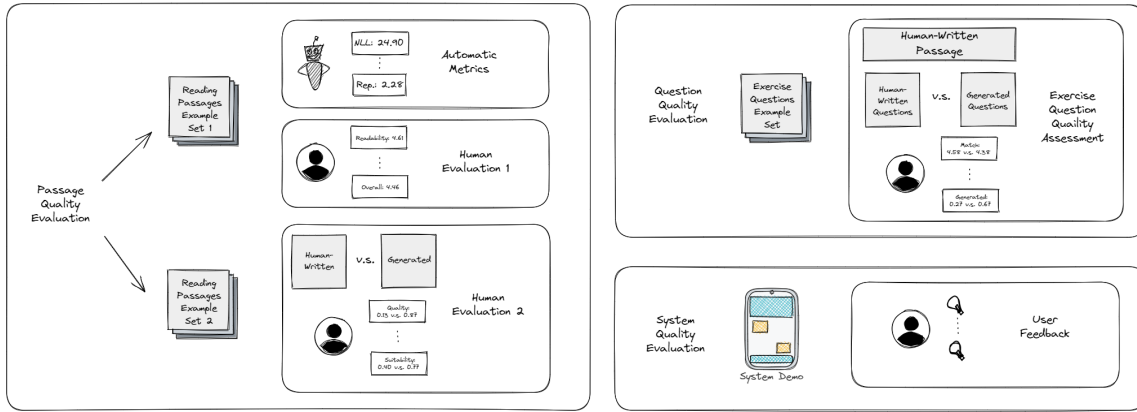


Figure 3: The illustration for each evaluation section.

20 sets of questions form the Exercise Questions Example Set, which is thoroughly evaluated in Section 4.2.

For the overall evaluation of our system (Section 4.3), we invite middle school educators, the intended users of our system, to utilize it first-hand. We request their insightful feedback and suggestions, furthering our goal of consistent improvement and customization to user needs.

#### 4.1 Reading Passage Quality Assessment

**Automatic Metrics** First, we apply automatic metrics commonly used in the literature on the Reading Passages Example Set 1. Table 1 presents the quantitative performance comparison of ChatGPT-generated reading passages with those produced by the baseline model and those written by human educators in textbooks. In general, the results indicate that the passages generated by the fine-tuned GPT-2 baseline are the easiest to read, and their average negative log-likelihood (NLL) is the lowest. However, this does not necessarily imply that the fine-tuned GPT-2 is the best model (Wang et al., 2022), as it may be overfitted in terms of NLL and generate text with high repetition. Moreover, high readability does not guarantee that the passages are logical and coherent, which are important dimensions for evaluating the quality of generated long text. The ChatGPT-generated passages receive the lowest readability scores, and also exhibit greater diversity.

In addition to automatic metrics, scores evaluated by experienced and trained human annotators serve as more reliable benchmarks (Clark et al., 2021). Next, we will introduce two designs for human evaluation in this study.

	Readability			Diversity	
	NLL	SMOG	Flesch	TTR	Rep.
Human	21.89	8.46	81.46	53.84	3.06
GPT-2	<b>18.60</b>	<b>6.59</b>	<b>92.50</b>	44.76	4.05
ChatGPT	24.90	9.81	73.29	<b>56.51</b>	<b>2.28</b>

Table 1: Results of automatic evaluation metrics on the Reading Passages Example Set 1. **NLL** (Alihosseini et al., 2019): the average negative log-likelihood loss; **SMOG** (McLaughlin, 1969): SMOG grade index estimates the years of education needed to understand the writing; **Flesch** (Flesch, 1979): Flesch reading-ease test, higher scores indicate material that is easier to read; **TTR (%)** (Richards, 1987; Celikyilmaz et al., 2020): the number of unique words (types) divided by the total number of words (tokens); **Rep. (%)** (Welleck et al., 2020; Pascual et al., 2021): the proportion of repeated 4-grams.

#### Human Evaluation 1: Multi-dimension Quality Scoring

We invite two groups of participants to assess the quality of the Reading Passages Example Set 1: 9 Chinese college students and 364 native English speakers. Chinese college students have years of English exercise training experience from middle school, and are familiar with reading comprehension exercises. Meanwhile, native English speakers possess a higher level of English proficiency than Chinese students, and their evaluation of the language may be more professional, but they have no idea what the reading passages in Chinese middle schools look like.

Before scoring each passage, the 9 student evaluators are given detailed guidelines about the evaluation rules, including the meanings of each quality dimension and two examples of middle school reading comprehension passages. To prevent fatigue, each evaluator is assigned only 30 passages. We

		Readability	Correctness	Coherence	Engagement	Overall Quality
Chinese Students	Human-Written	4.52	4.32	4.39	4.07	4.18
	Fine-tuned GPT-2	3.57	3.73	2.69	2.78	2.84
	ChatGPT (zero-shot)	<b>4.61</b>	<b>4.60</b>	<b>4.65</b>	<b>4.37</b>	<b>4.46</b>
Native Speakers	Human-Written	<b>3.79</b>	3.67	<b>3.77</b>	3.77	3.89
	Fine-tuned GPT-2	3.52	3.51	3.53	3.62	3.75
	ChatGPT (zero-shot)	3.78	<b>3.69</b>	<b>3.77</b>	<b>3.93</b>	<b>4.06</b>

Table 2: Quality scores of the three groups of passages in five dimensions evaluated by experienced Chinese students and English native speakers.

collect 270 individual evaluations in total, with 3 evaluations for each passage. For native English speakers, we recruit them from Amazon Mechanical Turk and collect 5 evaluations for each passage.

Each evaluation consists of 5 scores measuring different dimensions of text quality. These dimensions are widely used in human evaluations of text-generation studies and have been carefully selected based on their importance to the reading comprehension scenario. The explanations of quality dimensions are as follows:

- **Readability:** The extent to which texts are easy to read (Forrest et al., 2018; Di Fabrizio et al., 2014) and fluent (Mahapatra et al., 2016; Belz and Kow, 2010).
- **Correctness:** The extent to which texts accurately reflect facts and commonsense, how logical they are (Celikyilmaz et al., 2020), and whether they are proper in grammar (Wubben et al., 2016).
- **Coherence:** The extent to which texts are consistent with certain topics or storylines (Santhanam and Shaikh, 2019).
- **Engagement:** The extent to which texts are interesting and engaging.
- **Overall Quality:** The overall text quality of the reading passages.

The evaluation results are shown in Table 2. Surprisingly, as rated by experienced students, the quality scores of ChatGPT passages are higher than the scores of human-written passages across all selected dimensions. The passages generated by the fine-tuned GPT-2 baseline are generally of lower quality, and not comparable to the other two groups of passages. For the evaluations of native speakers, the scores of the passages are generally lower than those marked by Chinese students, since the reading materials used by middle school students may

be too simple for native speakers. Nonetheless, the conclusion does not change: ChatGPT passages have the highest overall quality.

We also conduct inter-annotator reliability tests to make sure the evaluation results are reliable. Among the student evaluators, we observe an average Pearson’s Correlation of 0.64, and the average Cronbach’s Alpha of the rating scores is 0.82, indicating a high internal consistency and a reliable measurement. Similar tests were conducted in the following human evaluations, all of which showed reliable results, so we will not elaborate on further.

## Human Evaluation 2: Pairwise Comparison

The three groups of generated passages (GPT-2 + PPLM, ChatGPT zero-shot, and ChatGPT one-shot generated) in the Reading Passages Example Set 2 are displayed side-by-side with human-written passages for evaluators to compare. In other words, each evaluator is presented with two passages at a time, one generated by the model and the other written by humans, with the order randomized.

We did not recruit native speakers for this evaluation but relied entirely on college students. Since we believe that native speakers who are not familiar with reading comprehension exercises in China are not suitable for the comparison evaluation. Another 9 students were recruited for Human Evaluation 2 to avoid the learning effect. Similar to Human Evaluation 1, we collect 3 evaluations for each set of passages. The evaluation questions are as follows.

- **Relative quality score.** Since the previous evaluation has already assessed multiple dimensions, here we only focus on the overall quality for simple verification. For the two passages displayed simultaneously, we ask the evaluators to mark the passage of better quality with a score of 1, and the other one with a score of 0. By taking the average at the level of passages and evaluators, we obtain three average quality scores for the three groups of generated passages and three for the human-written

ones, respectively. The following evaluation questions are analyzed in a similar way.

Table 3 shows that the ChatGPT scores are much higher than the baseline score. Moreover, evaluators believe that the quality of ChatGPT passages is even better than human-written ones (0.87 vs. 0.13 in the zero-shot setting and 0.80 vs. 0.20 in the one-shot setting), which is consistent with our findings in Human Evaluation 1. For the ChatGPT passages, the one-shot score is slightly lower than the zero-shot score (0.80 vs. 0.87), which may be due to more restrictions leading to a slight decrease in quality. Nonetheless, ChatGPT performs quite well in the reading passage generation task with our designed prompts.

	Human	Generated
Fine-tuned GPT-2 + PPLM	<b>0.70</b>	0.30
ChatGPT (zero-shot)	0.13	<b>0.87</b>
ChatGPT (one-shot)	0.20	<b>0.80</b>

Table 3: The comparison of **relative quality score** between human-written passages and generated ones. A higher score indicates better quality.

- **Model-Generated Score.** We also investigate whether evaluators can distinguish between passages written by humans and those generated by models. To do so, we design a simple Turing test by asking evaluators to assign a score of 1 if they believe the passage is generated by language models, and 0 otherwise. Therefore, the lower the score, the more likely the passage is perceived to be written by humans. From Table 4, we find that the passages generated by ChatGPT scored lower than the human-written passages displayed side-by-side, meaning that evaluators believe the ChatGPT passages are more likely to be human-written than the true ones, which is an interesting finding.

Another finding is that both generated and human-written passages in the one-shot setting scored the lowest. One plausible reason is that ChatGPT imitated the styles and structures of the referenced passage very well. When two similar passages of high quality appeared at the same time, evaluators tended to think that they were unlikely to be generated by models.

Note that if native speakers were asked to evaluate this dimension, the results might be different. Because they have a higher language proficiency and are more likely to notice characteristics that non-native speakers did not pay attention to.

- **Topic Coherence Score.** We examine whether

	Human	Generated
Fine-tuned GPT-2 + PPLM	<b>0.40</b>	0.57
ChatGPT (zero-shot)	0.53	<b>0.30</b>
ChatGPT (one-shot)	0.33	<b>0.23</b>

Table 4: The comparison of **model-generated score** between human-written passages and generated ones. A higher score indicates that the passage is more likely to be perceived from language models, instead of written by humans.

the passages are consistent with the given topics, that is, the control and personalization ability of the models. A score of 1 is given for consistency while 0 means inconsistency. Table 5 shows that even after fine-tuning with domain knowledge and with the extra control of PPLM, the GPT-2 baseline still did not generate passages that follow the given requirements well. In contrast, ChatGPT scored particularly high even in zero-shot (with a score of 0.97), indicating that it understands and follows the instructions specified in the prompts quite well.

	Human	Generated
Fine-tuned GPT-2 + PPLM	<b>0.87</b>	0.40
ChatGPT (zero-shot)	0.77	<b>0.97</b>
ChatGPT (one-shot)	0.77	<b>0.97</b>

Table 5: The comparison of **topic coherence score** between human-written passages and generated ones. A higher topic coherence score indicates that the passage is more consistent with the given topics.

- **Suitability Score.** This evaluation dimension requires the evaluator to have extensive experience with reading comprehension exercises and is not suitable for native English speakers who are unfamiliar with Chinese English education. If deemed suitable, the passage should receive a score of 1, 0 otherwise. Our findings in Table 6 show that evaluators generally believe that the passages generated by ChatGPT are largely suitable as reading comprehension materials and are even better than passages currently used as exercises.

	Human	Generated
Fine-tuned GPT-2 + PPLM	<b>0.53</b>	0.37
ChatGPT (zero-shot)	0.40	<b>0.77</b>
ChatGPT (one-shot)	0.53	<b>0.77</b>

Table 6: The comparison of **suitability score** between human-written passages and generated ones. A higher suitability score indicates that the passage is more suitable for middle school students in China.



In summary, the human evaluation results suggest that the ChatGPT passages generated by our system are of high quality across various dimensions, and even better than the human-written reading passages in many cases. The experienced evaluators believe that it is suitable to apply these materials in real educational contexts.

## 4.2 Exercise Question Quality Assessment

Next, we will evaluate the quality of the generated reading exercise questions. Currently, there is no reliable metric for evaluating the quality of generated multiple-choice questions, so we entirely rely on human evaluation.

Similar to how we evaluate passages in Human Evaluation 2, each evaluator is presented with two sets of questions, one generated by the system and one written by humans, along with the base passage in the Exercise Questions Example Set. The evaluators are asked to assess the quality of the questions according to various aspects, using scores ranging from 1 to 5. The following aspects are considered:

- **The extent to which the questions match the passage content.** We want to check whether the questions generated by our system align with the content of the passages and whether we can find correct answers within the passages. This is a basic requirement for the generated questions to be suitable for student practice.

- **The extent to which the questions are useful for the training of students.** Moreover, we ensure that the questions are not meaningless and that they can serve as effective exercises that contribute to students' English training.

- **The extent to which the questions are suitable for middle school English learners.** This dimension is similar to the previous one. Based on their extensive experience with English reading exercises, evaluators rate whether the generated questions are too difficult or too simple for students in Chinese middle schools.

- **The extent to which the questions appear to be written by language models.** If the generated questions exhibit certain patterns, they will be easily distinguished from the exercise questions in the textbook, indicating that the generated questions are too rigid and not flexible enough.

From Table 7, we observe that human-written questions outperform generated questions across all four dimensions. Although the generated questions are highly relevant to the passage content (with a

	Human	Generated
Match	<b>4.58</b>	4.38
Useful	<b>3.93</b>	3.25
Suitable	<b>3.92</b>	3.48
Generated or not	<b>0.27</b>	0.67

Table 7: The comparison of **exercise quality** in four dimensions between human-written and generated ones.

Match score of 4.38 out of 5), some of them exhibit obvious patterns, are too straightforward, and lack variation. Teachers may need to select suitable exercise questions from the various generated ones before assigning them to students.

## 4.3 System Quality Assessment

Our system, which integrates the features described above, is primarily designed for middle school teachers. To gather feedback on the system, we invited three experienced teachers in Beijing, who have many years of teaching experience, to personally use the system for a week and provide their feedback through interviews. Their feedback and suggestions are summarized in Table 12 in Appendix C.

Although there is still room for improvement, such as further optimizing the generation of multiple-choice questions, the quality of reading exercises generated by our system has greatly exceeded teachers' expectations. Teachers view this system as a valuable tool that can significantly reduce cost and time while providing students with more diverse and personalized learning materials.

## 5 Conclusion

In this study, we attempted to develop an educational system for teachers and English learners in Chinese middle schools that leverages the capabilities of LLMs to generate reading comprehension exercises. Extensive evaluations were conducted among various groups of representative human evaluators, and the high quality of the generated reading exercises was widely acknowledged. Experienced English teachers also provided extremely positive feedback on the system, indicating its potential for widespread use in real-world education. Our system is among the first applications of ChatGPT in educational contexts, and the valuable feedback and findings are likely to inspire future researchers and educators in integrating AI technology into education.

## Limitations

As noted in the evaluation section, our system does not perform perfectly in multiple-choice question generation, particularly when it comes to generating distracting options, even with the powerful ChatGPT. In the next step, we can adopt an open-source framework of LLMs and fine-tune a domain-specific model using the extensive educational materials provided by middle school teachers. This way, the question generation ability may be improved, and we will not need to rely on the OpenAI API.

On the other hand, although extensive evaluations have been conducted, they only involve a small fraction of teachers and students in a pre-interview setting. Once our system is widely deployed, a larger amount of user feedback will be collected and analyzed to monitor its effectiveness.

## References

- Danial Alihosseini, Ehsan Montahaei, and Mahdiah Soleymani Baghshah. 2019. [Jointly measuring diversity and quality in text generation models](#). In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 90–98, Minneapolis, Minnesota. Association for Computational Linguistics.
- David Baidoo-Anu and Leticia Owusu Ansah. 2023. Education in the era of generative artificial intelligence (ai): Understanding the potential benefits of chatgpt in promoting teaching and learning. *Available at SSRN 4337484*.
- Anja Belz and Eric Kow. 2010. [Comparing rating scales and preference judgements in language evaluation](#). In *Proceedings of the 6th International Natural Language Generation Conference*. 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*, pages 313–320, Trento, Italy. Association for Computational Linguistics.
- Anya 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, Dublin, Ireland. Association for Computational Linguistics.
- Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2007. [\(meta-\) evaluation of machine translation](#). In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 136–158, Prague, Czech Republic. Association for Computational Linguistics.
- Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey. *arXiv preprint arXiv:2006.14799*.
- Yu Chen, Scott Jensen, Leslie J Albert, Sambhav Gupta, and Terri Lee. 2023. Artificial intelligence (ai) student assistants in the classroom: Designing chatbots to support student success. *Information Systems Frontiers*, 25(1):161–182.
- 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, Online. Association for Computational Linguistics.
- Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. [Plug and play language models: A simple approach to controlled text generation](#). In *International Conference on Learning Representations*.
- Giuseppe Di Fabrizio, Amanda Stent, and Robert Gaizauskas. 2014. [A hybrid approach to multi-document summarization of opinions in reviews](#). In *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, pages 54–63, Philadelphia, Pennsylvania, U.S.A. Association for Computational Linguistics.
- Yogesh K Dwivedi, Nir Kshetri, Laurie Hughes, Emma Louise Slade, Anand Jeyaraj, Arpan Kumar Kar, Abdullah M Baabdullah, Alex Koohang, Vishnupriya Raghavan, Manju Ahuja, et al. 2023. “so what if chatgpt wrote it?” multidisciplinary perspectives on opportunities, challenges and implications of generative conversational ai for research, practice and policy. *International Journal of Information Management*, 71:102642.
- Mehmet Firat. 2023. How chat gpt can transform autodidactic experiences and open education. *Department of Distance Education, Open Education Faculty, Anadolu Unive*.
- Rudolf Flesch. 1979. How to write plain english. *University of Canterbury*.
- James Forrest, Somayajulu Sripada, Wei Pang, and George Coghill. 2018. [Towards making NLG a voice for interpretable machine learning](#). In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 177–182, Tilburg University, The Netherlands. Association for Computational Linguistics.

- Helen Hastie and Anja Belz. 2014. [A comparative evaluation methodology for NLG in interactive systems](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4004–4011, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Muhammad Khalifa, Hady Elsahar, and Marc Dymetman. 2021. [A distributional approach to controlled text generation](#). In *International Conference on Learning Representations*.
- 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*, 30:121–204.
- Junyi Li, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. [Pretrained language model for text generation: A survey](#). In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4492–4499. International Joint Conferences on Artificial Intelligence Organization. Survey Track.
- Joy Mahapatra, Sudip Kumar Naskar, and Sivaji Bandyopadhyay. 2016. Statistical natural language generation from tabular non-textual data. In *Proceedings of the 9th International Natural Language Generation conference*, pages 143–152.
- GH McLaughlin. 1969. Smog grading – a new readability formula. *Journal of Reading*, 12(8):639–646.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Damian Pascual, Beni Egressy, Clara Meister, Ryan Cotterell, and Roger Wattenhofer. 2021. [A plug-and-play method for controlled text generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3973–3997, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Tammy Pettinato Oltz. 2023. Chatgpt, professor of law. *Professor of Law (February 4, 2023)*.
- Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.
- 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.
- Brian Richards. 1987. Type/token ratios: What do they really tell us? *Journal of child language*, 14(2):201–209.
- Sashank Santhanam and Samira Shaikh. 2019. Towards best experiment design for evaluating dialogue system output. In *International Conference on Natural Language Generation*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Yequan Wang, Jiawen Deng, Aixin Sun, and Xuying Meng. 2022. Perplexity from plm is unreliable for evaluating text quality. *arXiv preprint arXiv:2210.05892*.
- Sean Welleck, Ilya Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. [Neural text generation with unlikelihood training](#). In *International Conference on Learning Representations*.
- Sander Wubben, Emiel Krahmer, Antal van den Bosch, and Suzan Verberne. 2016. [Abstractive compression of captions with attentive recurrent neural networks](#). In *Proceedings of the 9th International Natural Language Generation conference*, pages 41–50, Edinburgh, UK. Association for Computational Linguistics.
- Aeron Zentner. 2022. Applied innovation: Artificial intelligence in higher education. *Available at SSRN 4314180*.
- Xiaoming Zhai. 2022. Chatgpt user experience: Implications for education. *Available at SSRN 4312418*.
- Bo Zhang. 2023. [Preparing educators and students for chatgpt and ai technology in higher education](#).
- Chaoning Zhang, Chenshuang Zhang, Chenghao Li, Yu Qiao, Sheng Zheng, Sumit Kumar Dam, Mengchun Zhang, Jung Uk Kim, Seong Tae Kim, Jinwoo Choi, et al. 2023. One small step for generative ai, one giant leap for agi: A complete survey on chatgpt in aigc era. *arXiv preprint arXiv:2304.06488*.
- Ke Zhang and Ayse Begum Aslan. 2021. Ai technologies for education: Recent research & future directions. *Computers and Education: Artificial Intelligence*, 2:100025.
- Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texus: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100.

## A GPT-2 + PPLM Baseline

### A.1 Data

We collaborate with the Municipal Education Commission and 8 local middle schools in Beijing. We are provided 8,650 reading passages in total, including 5,066 supplemental reading materials (Dataset 1) and 3,584 currently used textbook exercise passages (Dataset 2), covering different difficulty levels from Grade 7 to Grade 9.

The descriptive statistics of our manually collected two datasets are shown in Table 8.

	Dataset 1	Dataset 2
# passages	5,066	3,584
min. length	32	30
avg. length	967.34	251.07
max. length	15,242	780

Table 8: Descriptive statistics of the two datasets.

Due to the confidentiality of educational resources, we are not able to publicly offer access to Dataset 1. Nonetheless, the trained model (with the fine-tuning process) using our datasets is provided in our GitHub repository.

### A.2 GPT-2 Fine-tuning

When fine-tuning, we adopt a two-step fine-tuning strategy to account for the different characteristics of the two datasets. In the first step, the model learns the general language features with a larger learning rate from Dataset 1. In the second step, fine-tuning on Dataset 2 with a lower learning rate and longer training epochs, the model is able to learn fine-grained characteristics of textbook reading passages, including formats, topics, and writing styles.

All the training processes of the GPT-2 baseline are implemented on a 16 GB NVIDIA Tesla P100 PCIe GPU provided by Google Colab.

We use the OpenAI GPT-2 medium model with 24-layer, 1,024-hidden layers, 16-heads, and 345M parameters, implemented by the Huggingface transformer library.

The textual materials in the dataset are tokenized by GPT-2 tokenizer. Since the max input length of the GPT-2 medium model is 1,024, we truncate all the passages that are longer than 1,024 tokens, and pad all passages that are shorter than 1,024 tokens to the same length of 1,024.

We randomly split the dataset into 80% as the training set and the remaining 20% as the test set.

The batch size is 2 and the random seed is 42. The AdamW optimizer with  $\epsilon = 10^{-8}$  is applied, and we adopt a linear learning schedule with 100 warm-up steps. The detailed training setting of our proposed two-step fine-tuning and other baseline strategies are shown in Table 9. The entire fine-tuning process using our two-step strategy takes approximately 6 hours.

	Learning rate	# epochs
Dataset 1	$1 \times 10^{-5}$	5
Dataset 2	$1 \times 10^{-5}$	3
Single-step	$1 \times 10^{-5}$	5
Two-step (1)	$5 \times 10^{-4}$	3
Two-step (2)	$1 \times 10^{-5}$	5

Table 9: Hyper-parameter setting for fine-tuning.

By manually examining the generated passages from all baseline strategies, we summarize and conclude that our two-step fine-tuning strategy achieves the best performance.

- **Fine-tuning with only dataset 1:** The lengths of generated passages are often too short or too long, and the word repetition problem often occurs.
- **Fine-tuning with only dataset 2:** The lengths of generated passages are often too short or too long. The format and word repetition problems exist.
- **Single-step fine-tuning with combined datasets:** The overall quality of the generated passages is higher than fine-tuning with only one dataset, but their length is still unstable.
- **Proposed two-step fine-tuning:** It performs the best, and the problems mentioned above are significantly alleviated.

### A.3 PPLM

To generate more coherent texts on a given topic, we apply a plug-and-play controllable text generation approach with topic keywords provided. It is expected that providing more keywords will lead to more coherent generated passages. We first provide a few (e.g., 3 to 5) initial topic words. This list can then be expanded to include more similar words (e.g., 30 words) by finding similar words based on word embeddings from a Word2Vec (Mikolov et al., 2013) model trained on our two reading datasets. Previous studies (Khalifa et al., 2021) showed that

PPLM tends to produce texts with frequent repetitions due to inappropriate hyper-parameters. Therefore, before applying PPLM to guide text generation, we use a simple grid search strategy to find the best hyper-parameters for each topic.

We adopt the Word2Vec model implemented by the gensim library<sup>5</sup> and train it from scratch with our reading passage datasets. The hyper-parameters of Word2Vec are as follows: *vector\_size=512*, *window=5*, *min\_count=5*, *workers=4*.

As mentioned above, a simple grid search is applied to seek the best hyper-parameters for each set of keywords, respectively. According to Dathathri et al. (2020), we tune the hyper-parameters that are relevant to the topic control intensity. The ranges of these parameters are listed in Table 10. The criterion to select hyper-parameters is based on manual examinations of the quality of generated passages. The set of hyper-parameters that guide the fine-tuned GPT-2 to generate passages with the highest overall quality will be regarded as the best one.

Parameter	Range
<i>step_size</i>	[0.02, 0.025, 0.03, 0.035, 0.04]
<i>gm_scale</i>	[0.7, 0.75, 0.8, 0.85, 0.9]
<i>kl_scale</i>	[0.01, 0.02, 0.03, 0.04, 0.05]
<i>grad_length</i>	[100, 1000, 10000]

Table 10: Grid search hyper-parameter bounds of PPLM.

## B Design of Reading Exercise Generation System

### B.1 Reading Passage Generation

**Zero-Shot setting** In the zero-shot setting, we instructed ChatGPT to be a helpful learning assistant capable of generating high-quality reading passages in the system prompt. We provided personalized requirements within the conversation prompt, including length, genre, difficulty, and topics. Reading passages for middle school students typically consist of around 200 words. Their difficulty level ranges from A1 to B2 according to the widely recognized CEFR standard, as middle school students are generally beginners. As for topics, teachers or students can freely select any

<sup>5</sup><https://radimrehurek.com/gensim/models/word2vec.html>

subject of interest using keywords, phrases, or sentences. ChatGPT’s remarkable ability enables it to comprehend these requirements and adhere to them throughout the text-generation process.

**One-Shot setting** In addition to creating reading passages from scratch, teachers often source content from the web or other materials and seek to adapt them into suitable reading passages for students. In the one-shot setting, we added an extra requirement: a referenced passage. Teachers can supply a referenced passage for ChatGPT, allowing the model to learn language styles and structural features. This setting facilitates more practical use of our system, though the added constraint may limit the model’s flexibility and creativity.

### B.2 Exercise Question Generation

We also generate questions and corresponding answer options for middle school reading comprehension exercises using appropriate prompts. Unlike the Q&A generation task in the NLP field, Chinese middle school students are mostly practicing multiple-choice selection questions. Few existing models focus on this task, and we have not identified a comparable method as a baseline for multiple-choice question generation. Given the high quality of ChatGPT-generated questions, we compare them directly to human-written exercise questions. For the prompt design, we input the number of questions, the number of options per question, and the question type for personalized customization. ChatGPT can generate exercise questions based on either a passage it previously created or a passage input by users. We did not set a difficulty level for the questions, as there is no reliable measurement standard. Nonetheless, question types can indirectly reflect difficulty. For example, logical inference questions are generally more challenging than word interpretation questions.

### B.3 Toxicity Check

To ensure the safety of middle school students and avoid ethical issues, we have implemented measures to prevent the generation of toxic text. In our prompts, we explicitly specify that the generated content must not contain violence, racism, or other harmful elements for young language learners. While OpenAI has devoted considerable attention to addressing toxicity concerns, and such texts are unlikely to appear in ChatGPT’s responses, we have implemented an additional layer of security

by using Google’s toxicity score tool<sup>6</sup> to screen the generated text. Exercises are made available to teachers and students only after passing the toxicity check.

#### **B.4 ChatGPT Prompts**

An example of the manually crafted prompts for the above tasks is presented in Table 11.

### **C Subjective Feedback from Users**

The evaluation and feedback from system users, that is, experienced middle school teachers, are summarized in Table 12.

### **D Examples of Generated Exercises**

Here we present several examples of human-written, GPT-2-generated, and ChatGPT-generated passages in Table 13. An example of a comparison between human-designed exercise questions and system-generated questions is shown in Table 14. You can also test our demo system to generate more reading comprehension exercises.

---

<sup>6</sup><https://developers.perspectiveapi.com/s/about-the-api-attributes-and-languages>

		Prompt
Passage	System	You are a helpful assistant to generate reading comprehension materials for Chinese middle school English learners. Your responses should not include any toxic content.
	Conversation (zero-shot)	Please generate a passage (without a title) that is similar to the given example and satisfies the following requirements: Topics: <i>{basketball competition}</i> ; Length: no more than <i>{200}</i> words; Genre: <i>{narrative}</i> ; CEFR level: <i>{B1}</i>
	Conversation (one-shot)	Please generate a passage (without a title) that is similar to the given example and satisfies the following requirements: Topics: <i>{basketball competition}</i> ; Length: no more than <i>{200}</i> words; Genre: <i>{narrative}</i> ; CEFR level: <i>{B1}</i> ; Example: <i>{a referenced passage}</i>
Question	System	You are a helpful assistant to generate reading comprehension exercise questions for Chinese middle school English learners. Your responses should not include any toxic content.
	Conversation	Please generate <i>{5}</i> multiple choice questions (each question with <i>{4}</i> choices), the corresponding answers and explanations for the following reading comprehension exercise. The type of questions should be <i>{inference}</i> questions. Exercise: <i>{input reading passage}</i>

Table 11: An example of the prompts for ChatGPT to generate high-quality reading comprehension exercises.

		Evaluations and Suggestions
Passages	Content	<ul style="list-style-type: none"> <li>✓ The generated passages are coherent in language.</li> <li>✓ The language characteristics are obvious and the quality of the generated passages is good.</li> </ul>
	Topic	<ul style="list-style-type: none"> <li>✓ The function of "generating based on the referenced passage" can present passages of different genres on the same topic effectively.</li> <li>✓ The system can perfectly follow the requirements of the topic, difficulty level, and passage genre.</li> </ul>
Exercises	Questions	<ul style="list-style-type: none"> <li>✓ The generated questions are of good quality and are based on the main idea and details of the passages.</li> <li>✓ Before using the system, I thought the AI can only generate exercise questions that are very simple and straightforward. Actually, the system can do more than that. The generated questions are usually good enough to help students understand the passages and examine their language ability.</li> <li>✗ The types of generated questions are not rich enough. It is easy to find their patterns, such as many of them are "What is something?", "What did someone do something?", "Why did someone do something?", etc.</li> </ul>
	Options	<ul style="list-style-type: none"> <li>✗ The quality of the questions is good, but the options are not so perfect. Some answer options are inaccurate or repetitive.</li> <li>✗ The correct answers are always accurate, but the wrong answers are of low quality. Sometimes they are too easy for students and cannot play a role as distractors.</li> </ul>
System	Usefulness	<ul style="list-style-type: none"> <li>✓ The system is like a personalized resource library. Rich information can be provided for teachers in daily teaching, which can further enhance teachers' ability to optimize resources while organizing them, thus providing diverse and personalized educational resources to improve students' English reading ability.</li> </ul>
	Ease of Use	<ul style="list-style-type: none"> <li>✓ The system interface is simple and the features are easy to understand.</li> <li>✓ It is easy to use the system even for teachers who know nothing about AI.</li> </ul>
	Overall Quality	<ul style="list-style-type: none"> <li>✓ I will rate the system 80/100. I am very satisfied with it.</li> <li>✓ This system is totally out of my expectation. I am happy such a powerful system will be applied in real-world education soon.</li> </ul>

Table 12: The evaluation of the system quality and the summarized feedback from experienced middle school teachers.

---

**Example Passages**

---

**Human-Written**

Deep in the forest lives David, who loves books. As soon as he reads one book, he brings home another. Soon the house is filled with books. His wife, Alice, is angry. You must do something! " David thinks hard and has an idea. "I can bring my books into the faraway hills and share what I have with the children. " Every week, David sets off across the country-side to faraway villages with his two donkeys. Early on a sunny morning, David and his donkeys stop at a river to drink. Then, a bandit jumps out! "Please let us pass, " David says. "The children are waiting. The bandit takes one book and shouts, "Next time I want some money! " They move on until at last, David sees some houses below. The children run to meet him. David reads them a story first. When the story ends, it's time for everyone to choose a book. The children hold their books close as they say good-bye and run home cheerfully. David and his donkeys head back, over and around the hills, and into the sunset. At home, Alice feeds her hungry husband and the donkeys. But then, instead of sleeping, David picks up his book, and reads deep into the night. And far away in the hills, candles and lanterns burn as the children read borrowed books deep into their night, too.

**Fine-tuned GPT-2**

One sunny afternoon, an old man sat down with his 25-year-old son on a bus. They were talking about some interesting things. The old man began to show his son the things he saw. The son asked, "Dad, why does your father not allow me to take off his bus? " The old man said, "Well, when our bus got full, I asked my son to sit on the floor and I sat down with him. He was very surprised. I asked him the other old man, "What do you mean, son? " The old man said, "I am just going to the park to play a game. The game is called 'Musical Chairs'. As we were playing, my son asked me, 'Dad, why does your son allow me to sit on the bus? '" "He doesn't want to get off the bus! " the son said with tears in his eyes. "Well, it's simple, but it's a difficult game for me. Every time I get off the bus, I have to sit down. If I don't get off, I will have to wait for the other driver to take me off. " The old man continued, "Now my son is very calm. We are going to the park to play. Can you play this game with me? " With tears in his eyes, the young man said, "Yes, I know! Look at my son. He was almost killed in a car accident two years ago. When my son got out of hospital, he had to wait for the other driver's help to get him back on the bus. "

**ChatGPT (zero-shot)**

Sarah was feeling a bit anxious as she entered the gym for her first dance class. She had always loved dancing, but had never taken a formal class before. As she looked around at the other students stretching and chatting, Sarah wondered if she would fit in. The class began with some warm-up exercises, and Sarah started to relax as she found that she could keep up with the others. They moved on to some basic steps, and Sarah felt her confidence growing with each new move. Before she knew it, the class was over, and Sarah couldn't believe how quickly the time had flown by. As she left the gym, she couldn't wait to come back for the next class. Reflecting on her experience, Sarah realized that trying new things could be scary, but it was important to take that first step. She also realized that she was capable of more than she had given herself credit for, and that the feeling of accomplishment was worth the initial discomfort. From that day forward, Sarah made a commitment to herself to try new things more often, and to approach each new experience with an open mind and a willingness to learn.

---

Table 13: Examples of human-written, GPT-2-generated, and ChatGPT-generated passages.



<b>Example</b>	
<b>Passage</b>	<p>Blue jackets, white shirts, and black pants... The school uniform looks so out of fashion. Like many of you, I don't like it. However, if I were given a chance, I wouldn't throw my school uniform away. Uniforms show that we are a part of an organization (our school). Wearing them shows we are all in this organization together. Some people argue that uniforms can improve learning. I completely agree with them. Uniforms make the classroom a serious environment. We will find it easier to focus on our schoolwork. Most importantly, a uniform means we don't have to worry about peer pressure. When everyone is dressed the same, we stop worrying about our appearance. There is no competition for being dressed in the latest fashion. This can save our parents a lot of money. In America, many schools do not have uniforms. Meanwhile, 160,000 children miss school every day, because of a fear of school bullying. This might not be directly connected with what the students are wearing. However, having a uniform can be safe for many students. It's hard to make fun of what someone is wearing when everyone is dressed the same. No one will feel left out because of what he is wearing. The school uniform gives us so much. They may not be fashionable, but that's just why I think they should be here to stay.</p>
<b>Human-Designed Exercise</b>	<p>(1) The writer doesn't like the school uniform because</p> <ul style="list-style-type: none"> <li>A. it's white.</li> <li>B. it's blue.</li> <li>C. it's out of fashion.</li> <li>D. it's old.</li> </ul> <p>(2) What does wearing uniforms show for students?</p> <ul style="list-style-type: none"> <li>A. They're too poor to buy their own clothes.</li> <li>B. They are a part of their school.</li> <li>C. They're good children of their parents.</li> <li>D. They like to look the same.</li> </ul> <p>(3) Which of the following is NOT true according to the third paragraph?</p> <ul style="list-style-type: none"> <li>A. The writer thinks wearing uniforms can make him/her nicer.</li> <li>B. The writer thinks wearing uniforms can make his/her classroom more serious.</li> <li>C. The writer thinks wearing uniforms can improve his/her study.</li> <li>D. The writer thinks wearing uniforms means we don't have to worry about peer pressure.</li> </ul> <p>(4) The passage mainly tells us</p> <ul style="list-style-type: none"> <li>A. wearing uniforms is important</li> <li>B. students must wear uniforms</li> <li>C. some schools don't have uniforms in America</li> <li>D. the uniforms are out of fashion</li> </ul>
<b>ChatGPT-Generated Exercise</b>	<p>(1) Why does the author wear their school uniform even though they do not like it?</p> <ul style="list-style-type: none"> <li>A. They do not have any other clothes to wear</li> <li>B. They believe it shows that they are part of their school community</li> <li>C. They think it is fashionable</li> <li>D. They want to fit in with their peers</li> </ul> <p>(2) What is one argument the author makes in favor of school uniforms?</p> <ul style="list-style-type: none"> <li>A. They can help students keep up with the latest fashion trends</li> <li>B. They can improve student learning</li> <li>C. They can be a source of peer pressure</li> <li>D. They can be expensive for parents to purchase</li> </ul> <p>(3) Why does the author suggest that school uniforms can be a good way to prevent bullying?</p> <ul style="list-style-type: none"> <li>A. Uniforms are fashionable and will make students popular</li> <li>B. Uniforms make it difficult to make fun of someone's clothing</li> <li>C. Uniforms are expensive and will make students feel more important</li> <li>D. Uniforms create a sense of competition among students</li> </ul> <p>(4) How do schools in America compare to schools in other countries with regards to school uniforms?</p> <ul style="list-style-type: none"> <li>A. American schools do not have uniforms, unlike many other countries</li> <li>B. American schools have uniforms, but they are not required</li> <li>C. American schools have uniforms, and they are required</li> <li>D. American schools have uniforms, but only for certain grades or classes</li> </ul>

Table 14: An example of a comparison between human-designed exercise questions and system-generated questions.

# Is ChatGPT a Good Teacher Coach?

## Measuring Zero-Shot Performance For Scoring and Providing Actionable Insights on Classroom Instruction

Rose Wang

rewang@cs.stanford.edu  
Stanford University

Dorottya Demszky

ddemszky@stanford.edu  
Stanford University

### Abstract

Coaching, which involves classroom observation and expert feedback, is a widespread and fundamental part of teacher training. However, the majority of teachers do not have access to consistent, high quality coaching due to limited resources and access to expertise. We explore whether generative AI could become a cost-effective complement to expert feedback by serving as an automated teacher coach. In doing so, we propose three teacher coaching tasks for generative AI: (A) scoring transcript segments based on classroom observation instruments, (B) identifying highlights and missed opportunities for good instructional strategies, and (C) providing actionable suggestions for eliciting more student reasoning. We recruit expert math teachers to evaluate the zero-shot performance of ChatGPT on each of these tasks for elementary math classroom transcripts. Our results reveal that ChatGPT generates responses that are relevant to improving instruction, but they are often not novel or insightful. For example, 82% of the model’s suggestions point to places in the transcript where the teacher is already implementing that suggestion. Our work highlights the challenges of producing insightful, novel and truthful feedback for teachers while paving the way for future research to address these obstacles and improve the capacity of generative AI to coach teachers.<sup>1</sup>

### 1 Introduction

Classroom observation, coupled with coaching, is the cornerstone of teacher education and professional development internationally (Adelman and Walker, 2003; Wragg, 2011; Martinez et al., 2016; Desimone and Pak, 2017). In the United States, teachers typically receive feedback from school administrators or instructional coaches, who assess teachers based on predetermined criteria and

rubrics. These structured evaluations often involve pre- and post-observation conferences, where the observer and teacher discuss teaching strategies and reflect on the observed instruction.

Despite its widespread adoption, classroom observation lacks consistency across schools and different learning contexts due to time and resource constraints, human subjectivity, and varying levels of expertise among observers (Kraft et al., 2018; Kelly et al., 2020). Frequency and quality of feedback can vary significantly from one school or learning context to another, resulting in disparities in teacher development opportunities and, consequently, student outcomes.

Prior work has sought to complement the limitations of manual classroom observation by leveraging natural language processing (NLP) to provide teachers with scalable, automated feedback on instructional practice (Demszky et al., 2023a; Suresh et al., 2021). These approaches offer low-level statistics of instruction, such as the frequency of teaching strategies employed in the classroom—different from the high-level, actionable feedback provided during coaching practice. Receiving high-level, actionable feedback automatically could be easier for teachers to interpret than low level statistics, and such feedback also aligns more closely with existing forms of coaching.

Recent advances in NLP have resulted in models like ChatGPT that have remarkable few-shot and zero-shot abilities. ChatGPT has been applied to various NLP tasks relevant to education, such as essay writing (Basic et al., 2023) or assisting on mathematics problems (Pardos and Bhandari, 2023), and providing essay feedback to students (Dai et al., 2023). A survey conducted by the Walton Family Foundation shows that 40% of teachers use ChatGPT on a weekly basis for tasks such as lesson planning and building background knowledge for lessons (Walton Family Foundation, 2023). Given ChatGPT’s potential and teachers’ growing

<sup>1</sup>The code and model outputs are open-sourced here: <https://github.com/rosewang2008/zero-shot-teacher-feedback>.

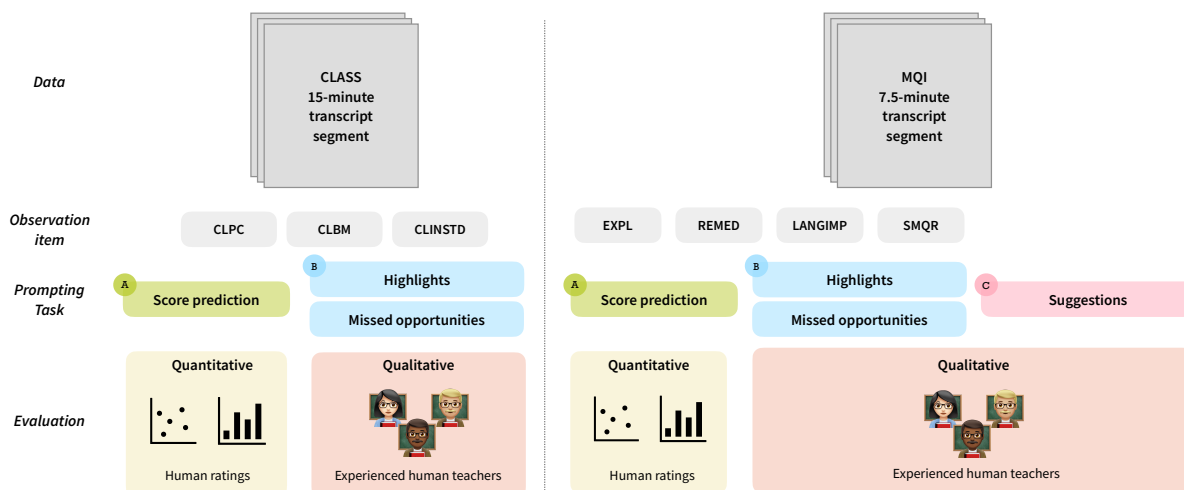


Figure 1: Setup for the automated feedback task. Our work proposes three teacher coaching tasks. Task A is to score a transcript segment for items derived from classroom observation instruments; for instance, CLPC, CLBM, and CLINSTD are CLASS observation items, and EXPL, REMED, LANGIMP, SMQR are MQI observation items. Task B is to identify highlights and missed opportunities for good instructional strategies. Task C is to provide actionable suggestions for eliciting more student reasoning.

familiarity with it, we are interested in the following research question: Can ChatGPT help instructional coaches and teachers by providing effective feedback, like generating classroom observation rubric scores and helpful pedagogical suggestions?

To answer this question, we propose the following teacher coaching tasks for generative AI.

**Task A.** *Score* a transcript segment for items derived from classroom observation instruments

**Task B.** *Identify highlights and missed opportunities* for good instructional strategies

**Task C.** *Provide actionable suggestions* for eliciting more student reasoning

We evaluate the performance of ChatGPT with zero-shot prompting on each of these tasks via the process in Figure 1. We use the NCTE dataset (Demszky and Hill, 2022), a large dataset of elementary math classroom transcripts. The data is annotated by experts with two observation protocols: the Classroom Assessment Scoring System (CLASS) (Pianta et al., 2008) and Mathematical Quality Instruction (MQI) (Hill et al., 2008) instruments. We prompt ChatGPT to score segments from these transcripts (Task A) and to identify highlights and missed opportunities (Task B) with respect to items derived from CLASS and MQI. Finally, we prompt the model to generate suggestions

to the teacher for eliciting more student mathematical reasoning in the classroom (Task C). We evaluate ChatGPT by comparing the model’s numerical predictions to raters’ scores in the NCTE data (Task A). We also recruit math teachers to rate the ChatGPT’s responses along multiple helpfulness criteria (Tasks B & C).

We find that ChatGPT has significant room for improvement in all three tasks, but still holds promise for providing scalable high-quality feedback. On predicting scores, ChatGPT has low correlation with human ratings across all observation items even with added rubric information and reasoning. On identifying highlights and missed opportunities, ChatGPT generates responses that are often not insightful (50-70%) or relevant (35-50%) to what is being asked for by both instruments. Finally, the majority of suggestions generated by ChatGPT (82%) describe what the teacher already does in the transcript. Nonetheless, the model does generate a majority of suggestions that are actionable and faithfully interpret the teaching context. We believe that with further development, ChatGPT can become a valuable tool for instructional coaches and teachers. Our work highlights an exciting area for future research to improve on the current limitations of automated feedback systems.

In sum, we make the following contributions: we (1) propose three teacher coaching tasks for

generative AI, (2) recruit expert teachers to evaluate ChatGPT’s zero-shot performance on these tasks given elementary math classroom transcripts, (3) demonstrate that ChatGPT is useful in some aspects but still has a lot of room for improvement, and finally (4) highlight directions for future directions towards providing useful feedback to teachers.

## 2 Related Work

**Automated feedback to educators.** Prior works on automated feedback tools provide analytics on student engagement and progress (Su et al., 2014; Schwarz et al., 2018; Aslan et al., 2019; Bonneton-Botté et al., 2020; Alrajhi et al., 2021, among others). These tools enable teachers to monitor student learning and intervene as needed. Recent NLP advances are able to provide teachers feedback on their classroom discourse, promoting self-reflection and instructional development (Samei et al., 2014; Donnelly et al., 2017; Kelly et al., 2018; Jensen et al., 2020). For example, Suresh et al. (2021) provides feedback to teachers on their teaching moves, such as how frequently the teacher revoices a student’s idea or how frequently the teacher asks students to reason aloud. Jacobs et al. (2022) provides evidence that K-12 math teachers receive this kind of feedback positively. A similar tool, M-Powering Teachers, provides feedback to teachers on their uptake of student ideas and demonstrates effectiveness in the 1-on-1 learning setting (Demszky and Liu, 2023). and online group instruction Demeszky et al. (2023b). Altogether, these findings show a positive impact of cost-effective automated tools. They prompt further investigations into what other types of automated feedback are effective. Our work constitutes one exploration in this area.

**Testing zero-shot capabilities of ChatGPT.** Recent works have measured the capabilities of ChatGPT for annotation on established datasets and benchmarks (Kuzman et al., 2023; He et al., 2023; Gilardi et al., 2023; Dai et al., 2023). For example, in a non-education setting, Gilardi et al. (2023) evaluates the zero-shot ability of ChatGPT to classify tweets. Dai et al. (2023) is a recent education work that investigates ChatGPT’s zero-shot ability to provide feedback to students on business project proposals. However, their study only utilizes a single broad prompt to solicit feedback and they do not evaluate for common model issues like hallucination (Ji et al., 2023). Our work proposes

three concrete tasks to generate different forms of feedback for teachers, and our evaluation targets common qualitative issues in model generations. For other recent applications of ChatGPT, we refer the reader to Liu et al. (2023).

## 3 Data

We use the National Center for Teacher Effectiveness (NCTE) Transcript dataset (Demszky and Hill, 2022) in this work, which is the largest publicly available dataset of U.S. classroom transcripts linked with classroom observation scores. The dataset consists of 1,660 45-60 minute long 4th and 5th grade elementary mathematics observations collected by the NCTE between 2010-2013. The transcripts are anonymized and represent data from 317 teachers across 4 school districts that serve largely historically marginalized students.

Transcripts are derived from video recordings, which were scored by expert raters using two instruments at the time of the NCTE data collection: the Classroom Assessment Scoring System (CLASS) (Pianta et al., 2008) and Mathematical Quality Instruction (MQI) (Hill et al., 2008) instruments. We evaluate ChatGPT’s ability to predict scores for both instruments, as described below.

**The CLASS instrument.** CLASS is an observational instrument that assesses classroom quality in PK-12 classrooms along three main dimensions: *Emotional Support*, *Classroom Organization* and *Instructional Support*. Each of these dimensions is measured by multiple observation items; we choose one item from each dimension to provide a proof-of-concept. For *Emotional Support*, we focus on the POSITIVE CLIMATE (CLPC) item, which measures the enjoyment and emotional connection that teachers have with students and that students have with their peers. For *Classroom Organization*, we focus on the BEHAVIOR MANAGEMENT (CLBM) item which measures how well the teachers encourage positive behaviors and monitor, prevent and redirect misbehavior. Finally, for *Instructional Support*, we focus on the INSTRUCTIONAL DIALOGUE (CLINSTD) dimension which measures how the teacher uses structured, cumulative questioning and discussion to guide and prompt students’ understanding of content. Each item is scored on a scale of 1-7 where 1 is low and 7 is high. All items are scored on a 15-minute transcript segment, which is typically about a third or fourth of the full classroom duration.

**The MQI instrument.** The MQI observation instrument assesses the mathematical quality of instruction, characterizing the rigor and richness of the mathematics in the lesson, along four dimensions: *Richness of the Mathematics*, *Working with Students and Mathematics*, *Errors and Imprecision*, and *Student Participation in Meaning-Making and Reasoning*. Similar to CLASS, each of these dimensions is measured by several observation items and we select one from each. For *Richness of the Mathematics*, we focus on the EXPLANATIONS (EXPL) dimension which evaluates the quality of the teacher’s mathematical explanations. For *Working with Students and Mathematics*, we focus on the REMEDIATION OF STUDENT ERRORS AND DIFFICULTIES (REMED) which measures how well the teacher remediates student errors and difficulties. For *Errors and Imprecision*, we focus on the IMPRECISION IN LANGUAGE OR NOTATION (LANGIMP) dimension which measures the teacher’s lack of precision in mathematical language or notation. Finally, for *Student Participation in Meaning-Making and Reasoning*, we focus on the STUDENT MATHEMATICAL QUESTIONING AND REASONING (SMQR) dimension which measures how well students engage in mathematical thinking. These items are scored on scale of 1-3 where 1 is low and 3 is high. They are scored on a 7.5 minute transcript segment, which is typically a seventh or eighth of the full classroom duration.

### 3.1 Pre-processing

**Transcript selection.** Due to classroom noise and far-field audio, student talk often contains inaudible talk marked as “[inaudible]”. In preliminary experiments, we notice that ChatGPT often overinterprets classroom events when “[inaudible]” is present in the student’s transcription. For example, the model misinterprets the transcription line “student: [inaudible]” as “A student’s response is inaudible, which may make them feel ignored or unimportant.” or the line “Fudge, banana, vanilla, strawberry, banana, vanilla, banana, [inaudible]. [...]” as the teacher allowing students to talk over each other and interrupt the lesson. To reduce the occurrences of the model overinterpreting the classroom events and best evaluate the model’s ability to provide feedback, we only consider transcripts where less than 10% of the student contributions includes an “[inaudible]” marker. Because these transcripts are very long and it would be costly to

evaluate ChatGPT on all of the transcripts, we randomly pick 10 for the CLASS instrument and 10 for the MQI instrument to use.

**Transcript segmentation.** The CLASS observation instrument applies to 15-minute segments and MQI to 7.5-minute segments. Each transcript has an annotation of the total number of CLASS segments and MQI segments. We split each transcript into segments by grouping utterances into equal-sized bins. For example, if a transcript has 3 CLASS segments and 300 utterances, we each segment will have 100 utterances each.

**Segment formatting.** In the *quantitative* Task A experiments, every utterance in the transcript segment is formatted as: “<speaker>: <utterance>”. <speaker> is either the teacher or a student and <utterance> is the speaker’s utterance. In our *qualitative* Task B and C experiments, we mark every utterance with a number. The utterance is formatted as: “<utterance number>. <speaker>: <utterance>”. We use utterance numbers in the qualitative experiments because our prompts ask the model to identify utterances when providing specific feedback. In contrast, the quantitative experiments evaluate the entire transcript segment holistically.

## 4 Methods

We use the `gpt-3.5-turbo` model through the OpenAI API, the model that powers ChatGPT. We decode with temperature 0. We employ zero-shot prompting in our study for three reasons. First, transcript segments are long, and the length of annotated example segments would exceed the maximum input size. Second, zero-shot prompting mimics most closely the current ways in which teachers interact with ChatGPT. Third, we are interested in evaluating ChatGPT’s capabilities off-the-shelf, without additional tuning.

### 4.1 Prompting

We provide an overview of prompting methods. Appendix A contains all the prompts used in this work and information about how they are sourced.

**Task A: Scoring transcripts.** We zero-shot prompt ChatGPT to predict observation scores according to the CLASS and MQI rubrics. We employ three prompting techniques: (1) prompting to directly predict a score with 1-2 sentence summary of the item (*direct answer*, DA) – see example for

CLBM in Figure 6, (2) same as DA but with additional one-sentence descriptions for low/mid/high ratings (*direct answer with description*, DA<sup>+</sup>) and (3) same as DA, with asking the model to provide reasoning before predicting a score (*reasoning then answer*, RA). RA follows recent literature on LLM prompting with reasoning where models benefit from added reasoning on mathematical domains (Wei et al., 2022, *inter alia*). The item descriptions all derived from the original observation manuals, condensed to fit the context window of the model while accounting for space taken up by the transcript segment. For all the prompts, the model correctly outputs integer values within each observation instrument’s score range.

**Task B: Identify highlights and missed opportunities.** We zero-shot prompt ChatGPT to identify and elaborate on highlights and missed opportunities for CLASS and MQI items. Specifically, we prompt ChatGPT to identify 5 good and bad examples (i.e. missed opportunities or poor execution) of each dimension. The prompt includes numbered transcript sentences and asks the model to indicate the line number, before explaining the example. See Figure 2 for an example of the prompt and model outputs.

**Task C: Provide actionable suggestions for eliciting student reasoning.** We zero-shot prompt ChatGPT to provide actionable suggestions to the teacher for eliciting more student mathematical reasoning in the classroom. This task is open-ended: We simply ask the model to provide 5 suggestions for the teacher for eliciting more student reasoning in the classroom. We provide the model with a definition of student reasoning. The prompt includes numbered transcript sentences and asks the model to indicate the line number, before explaining the example. We use the MQI transcript segments because they are shorter and more likely to fit within the model’s context window. See Figure 3 for an example of the prompt and model outputs.

## 4.2 Validation

We describe the analytical methods we use to answer each of the research questions.

**Task A.** The NCTE transcript dataset contains CLASS and MQI scores from human annotators. We compare ChatGPT’s predictions against the human annotator scores. We randomly pick 100 transcript segments from the processed dataset (rf.

### Example of Task B

Consider the following classroom transcript.

Transcript:

1. student: Because -
2. teacher: Nice and loud so we can hear you.
3. student: I did 12 times 3 equals 36.
4. teacher: Okay, 12 times 3 equals 36.
5. student: Times 2, 36 times 2.
6. teacher: Then 36 times 2 equals 72. And that’s your answer, right? Seventy-two inches. Where’d you get the 12 from?  
[...]
26. teacher: We’re doing something different now. I need everyone’s attention up front. Everyone should be looking up here. Paying attention to me. Right now the question is right here. How many inches are in one yard?  
[...]

Please do the following.

1. Provide up to 5 good examples of the teacher’s behavior management. Behavior management encompasses the teacher’s use of effective methods to encourage desirable behavior and prevent and re-direct misbehavior.
2. Provide up to 5 bad examples (eg. missed opportunities or poor execution) of the teacher’s behavior management.  
[...]

### Model response

Good examples:

1. Line number: 2, Segment: ‘Nice and loud so we can hear you.’, Reason: The teacher encourages the student to speak up and be heard.  
[...]

Bad examples:

2. Line number: 26, Segment: ‘We’re doing something different now.’, Reason: The teacher abruptly changes the topic without providing any context or explanation.  
[...]

Figure 2: The top box shows an example of task B for identifying highlights and missed opportunities on the teacher’s behavior management (CLBM). The bottom box shows the model’s output. For space reasons, the prompt and generations are abbreviated in this example, indicated by [...].

Section 3.1) for evaluation. We compute Spearman correlation to measure how well the model’s predictions correspond to the human ratings. We also compare the distribution of human ratings vs model ratings via a histogram, to understand how

### Example of Task C

Consider the following classroom transcript.

Transcript:

1. student: Divisor.
2. teacher: Divided. It means the numerator divided by the denominator. That's what that line means. We also know that when we see, when something's in division it also looks like this. That means divided by, but here it's just the line. Then we see numbers right across the top here. Let's read these numbers with me everybody. Can everybody see? What are they? 12.
3. teacher: Good. Then along the sides we see some more numbers. We've been forming one, two, three, four, five, six, seven, eight, nine, 10, 11, 12. So the numbers across the top are called the what? Numerators. The numbers down the side are called what? Denominators. So this is almost going to be like a division table. [...]

The transcript contains many short student responses. Please provide 5 suggestions for the teacher on how the teacher could elicit more student reasoning in the classroom. Student reasoning is counted broadly as students asking questions, engaging in mathematical discourse with their teacher or peers, and providing explanations such as justifying their answers. [...]

Advice to the teacher:

### Example of model output

1. Line number: 2, Segment: "Let's read these numbers with me everybody. Can everybody see? What are they? 12.", Suggestion: *Instead of simply asking students to read the numbers, ask them to explain how they know the numbers are 12 and to provide examples of other numbers they see in the table.* [...]

Figure 3: Example on task C for providing suggestions on eliciting more student mathematical reasoning in the classroom. The model's output is italicized. For space reasons, the prompt and generations are abbreviated in this example, indicated by [...].

well ChatGPT is calibrated for this task.

**Task B.** We randomly pick 10 transcript segments and prompt the model to identify highlights and missed opportunities per observation item in CLASS and MQI. We randomly select two high-

lights and two missed opportunities to be evaluated.

This results in 216 CLASS examples (= 18 segments  $\times$  3 CLASS codes  $\times$  (2 highlights + 2 missed opportunities)) and 288 MQI examples (= 18 segments  $\times$  4 MQI codes  $\times$  (2 highlights + 2 missed opportunities)). We recruit two math teachers to evaluate the model's outputs: one of the teachers has decades of experience as an instructional coach, and the other has 6 years of math teaching experience in title 1 public schools. Examples were split evenly between the teachers.

Teacher are asked to rate each example along three criteria, which we identify based on preliminary experiments (e.g. observed hallucination) and by consulting the teachers.

1. *Relevance*: Is the model's response relevant to the CLASS or MQI item of interest?
2. *Faithfulness*: Does the model's response have an accurate interpretation of the events that occur in the classroom transcript?
3. *Insightfulness*: Does the model's response reveal insights beyond a literal restatement of what happens in the transcript?

Each criteria is evaluated on a 3-point scale (yes, somewhat, no) with optional comments. For more details on the experimental setup and interrater comparison, please refer to Appendix B.

**Task C.** We evaluate this task similarly to Task B, except for slight changes in the criteria. We prompt the model using the 18 transcript segments from Task B to generate suggestions for eliciting more student reasoning. We randomly sample 2 suggestions per segment, resulting in 36 examples. Examples were split evenly between annotators. We use the following evaluation criteria:

1. *Relevance*: Is the model's response relevant to eliciting more student reasoning?
2. *Faithfulness*: Does the model's response have the right interpretation of the events that occur in the classroom transcript?
3. *Actionability*: Is the model's suggestion something that the teacher can easily translate into practice for improving their teaching or encouraging student mathematical reasoning?
4. *Novelty*: Is the model suggesting something that the teacher already does or is it a novel suggestion? Note that the experimental interface asks

about “redundancy”; we reverse the rating here for consistency across criteria (higher= better).

Similar to the previous section, we ask the teachers to evaluate on a 3-point scale (yes, somewhat, no) with optional comments.

## 5 Results & Discussion

	CLPC	CLBM	CLINSTD
DA	0.00	0.35	-0.01
DA <sup>+</sup>	0.04	0.23	0.07
RA	-0.06	0.07	-0.05

	EXPL	REMED	LANGIMP	SMQR
DA	0.02	0.05	0.00	0.17
DA <sup>+</sup>	0.12	0.06	0.02	0.17
RA	-0.11	-0.06	0.04	0.06

Table 1: The Spearman correlation values between the human scores and model predictions on the CLASS dimensions (top table) and MQI dimensions (bottom table). The columns represent the different dimensions and the rows represent the different prompting methods discussed in Section 4.

**Task A: Scoring transcripts.** ChatGPT performs poorly at scoring transcripts both for MQI and CLASS items. Table 1 reports the Spearman correlation values, and Figure 4 reports the score distributions. Appendix C contains additional plots, including a comparison of the human vs. model score distributions.

As for CLASS, two findings are consistent across our prompting methods. First, the model tends to predict higher values on all CLASS dimensions than human ratings and it performs best on CLBM. We hypothesize that CLBM may be easier to predict because (i) it is the only item whose distribution is skewed towards higher values and (ii) because scoring behavior management requires the least pedagogical expertise. Interestingly, adding more information to the prompt like per-score descriptions (DA<sup>+</sup>) or allowing for reasoning (RA) did not improve the correlation score—in some cases making the score worse, such as for CLBM.

As for MQI, for all dimensions but REMED the model tends to predict the middle score (2 out of 3); this observation is consistent across all prompting methods. Another interpretation of this finding, consistent with the CLASS results (which is on a 7 point scale), is that the model tends to predict the

second to highest rating. We do not have sufficient data to disentangle these two interpretations.

For REMED, the model generally predicts the highest rating (Figure 4). Similar to the observations made in CLASS, adding more information or reasoning does not help the model. The model seems to pick up on SMQR better than the other items, but its correlation decreases with both added information and reasoning.

Altogether, the models’ tendency to predict the same scores for the same MQI or CLASS item suggest that the predicted scores are a function of the dimension description and not of the transcript evidence or the prompting methodology.

**Task B: Identify highlights and missed opportunities.** Figure 5a summarizes the ratings on model responses for the CLASS instrument, and Figure 5b for the MQI instrument. Teachers generally did not find the model responses insightful or relevant to what was being asked for both instruments. Hallucination, as rated by *faithfulness*, is not the most problematic dimension out of the three. Nonetheless, it appears in a nontrivial amount of the model responses—around 20-30% of the model responses are marked with being unfaithful in interpreting the classroom transcript.

Interestingly, the MQI results are worse than the CLASS results across all evaluation dimensions. Concretely, the “No” proportions increase on every dimension from CLASS→MQI: Low scores on *faithful* increase 22 → 29% (+7), *relevant* 35 → 55% (+20), and *insightful* 51 → 71% (+20). This suggests that the model performs relatively worse on interpreting and evaluating technical aspects of math instruction quality. Appendix C contains additional plots, including the Cohen’s kappa between raters.

**Task C: Provide actionable suggestions for eliciting student reasoning.** Figure 5c summarizes the ratings on the model suggestions. The most noticeable observation is that the model tends to produce redundant suggestions (opposite of *novelty*), repeating what the teacher already does in the transcript 82% of the time. Nonetheless, most model responses were rated to be *faithful* to the transcript context, *relevant* to eliciting more student reasoning, and *actionable* for the teacher to implement.

The results for Task B and C may be explained by the fact that ChatGPT was unlikely to see exam-



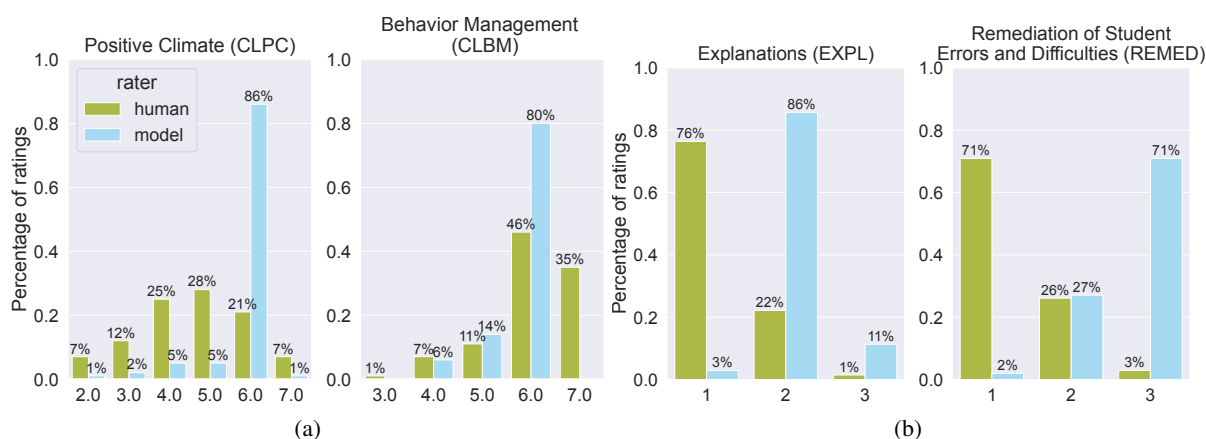


Figure 4: **Human and model distribution over scores for CLASS and MQI (Task A).** The model scores are collected using DA prompting on (a) CLPC and CLBM, and (b) EXPL and SMQR.

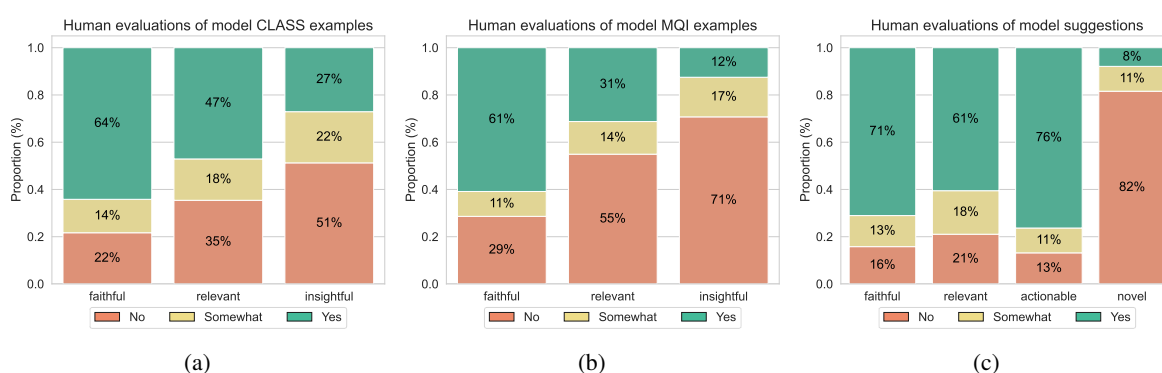


Figure 5: Math teachers' evaluations for (a) highlights and missed opportunities (Task B) on CLASS items, (b) highlights and missed opportunities (Task B) on MQI items and (c) suggestions for eliciting more student reasoning (Task C).

ples of instructional feedback, let alone examples of teacher coaching during its training, given the scarcity of publicly available data in this area. Thus, it has only learned to reproduce patterns already observed in the text, and not to produce out-of-the-box expert suggestions.

## 6 Limitations

This section discusses the limitations related to the evaluation process and potential ethical considerations associated with the use of ChatGPT or similar language models in educational settings.

**Human evaluation** Our evaluation is conducted with a limited sample size of two teachers. Future work should aim to include a larger and diverse sample of teachers to capture a wider range of perspectives. This would help tease apart the potential teacher biases from generalizable claims about the feedback quality.

**Ethical considerations** The use of language models like ChatGPT in educational contexts war-

rants careful examination. For example, because the model relies on transcribed speech and is trained on primarily English, it might misinterpret the transcriptions of teachers or students who do not speak English fluently. Additionally, deploying language models in education settings raises concerns regarding privacy and data security. For example, the raw classroom transcripts should not be directly fed into the model to provide feedback as it may contain personally identifiable information about students. Guardrails should be set to prevent classroom data from being sent directly to external companies.

## 7 Avenues for Future Work

As evidenced from our work, generating good feedback for teaching is *challenging* and ChatGPT has significant room for improvement in this area. This section discusses potential future directions to overcome these obstacles.

**Reducing hallucination.** Our results show that ChatGPT does generate a non-trivial amount of misleading responses as measured by our faithfulness dimension (15-30% of the time). This observation is documented in the LLM literature as model hallucination (Ji et al., 2023). In domains that leverage references or citations such as in fact-checking, remedies include retrieving sources and checking the claims made by the model (Nakano et al., 2022; Menick et al., 2022, *inter alia*). In the domain of teacher feedback, however, it is not obvious what the “true” interpretation is, as even human observers may disagree slightly with respect to the teachers’ intentions or actions. Future work could decrease hallucination in these higher inference domains, e.g. by forcing the model to be conservative with respect to making inferences.

**Involving coaches and educators in model tuning.** Our results show that ChatGPT struggles to generate insightful and novel feedback for teachers; understandably, since such feedback is not present in its training data. Involving coaches and educators in the reinforcement learning stage of model fine-tuning (Christiano et al., 2017) could be an effective way to improve the models’ performance for teacher coaching. One less costly alternative is to engineer the model’s prompt collaboratively with teachers and coaches. However, we are sceptical about the effectiveness of prompt engineering for teacher feedback, as it does not address model’s lack of exposure to teacher coaching examples during training.

**Tailoring feedback to a teacher’s needs and expanding to other subjects.** What counts as helpful feedback may be different for each teacher, and look different in other subjects, eg. History and English. Even for the same teacher, what they *self-report* to be helpful may be different from what what has a positive *impact* on their practice. An effective coach takes this into account, and is able to dynamically adapt the feedback based on the teacher’s needs and based on what they observe to be effective for that teacher (Thomas et al., 2015; Kraft and Blazar, 2018). Improving ChatGPT’s ability to differentiate feedback based on the teacher’s needs, and update the feedback strategy based on teacher’s subsequently observed practice would be a valuable direction for future work.

To adapt our approach beyond mathematics, such as in subjects like History or English, re-

searchers and instructors should collaborate and account for the subject’s instructional practices and learning objectives. This would help identify the relevant dimensions of effective teaching and inform the design of feedback prompts. For example, they can build on the subject-specific observation instruments as done in our work.

**Integrating automated feedback into human coaching practice.** We envision automated coaching to complement, rather than replace coaching by experts for three reasons. First, as this paper shows, the capabilities of current technology is very far from that of an expert instructional coach. Second, even with improved technology, having an expert in the loop mitigates the risks of misleading or biased model outputs. Finally, even though automated feedback offers several benefits, including flexibility, scalability, privacy, lack of judgment, human interaction is still an important component of coaching and is perceived by teachers as such (Hunt et al., 2021). Automated coaching could complement human coaching in a *teacher-facing* way, e.g. by directly providing the teacher with feedback on-demand. Such an automated tool can also be *coach-facing*, e.g. by generating diverse range of suggestions that the coach can then choose from based on what they think is most helpful for the teacher they are supporting.

## 8 Conclusion

Our work presents a step towards leveraging generative AI to complement the limitations of manual classroom observation and provide scalable, automated feedback on instructional practice. While our results reveal that ChatGPT has room for improvement in generating insightful and novel feedback for teaching, our proposed tasks and evaluation process provide a foundation for future research to address the challenges of teacher coaching using NLP. Our work underscores the challenge and importance of generating *helpful* feedback for teacher coaching. Moving forward, we propose several directions for further research, such as improved prompting methods and reinforcement learning with feedback from coaches. Ultimately, we envision a future where generative AI can play a crucial role in supporting effective teacher education and professional development, leading to improved outcomes for students.

## Acknowledgements

REW is supported by the National Science Foundation Graduate Research Fellowship. We thank Jiang Wu and Christine Kuzdzal for their helpful feedback.

## References

- Clement Adelman and Roy Walker. 2003. *A guide to classroom observation*. Routledge.
- L. Alrajhi, A. Alamri, F. D. Pereira, and A. I. Cristea. 2021. Urgency analysis of learners' comments: An automated intervention priority model for mooc. In *International Conference on Intelligent Tutoring Systems*, pages 148–160.
- S. Aslan, N. Alyuz, C. Tanriover, S. E. Mete, E. Okur, S. K. D'Mello, and A. Arslan Esme. 2019. Investigating the impact of a real-time. In *Multimodal student engagement analytics technology in authentic classrooms*, pages 1–12. of the 2019 CHI conference on human factors in computing systems.
- Zeljana Basic, Ana Banovac, Ivana Kruzic, and Ivan Jerkovic. 2023. [Better by you, better than me, chatgpt3 as writing assistance in students essays](#).
- Nathalie Bonneton-Botté, Sylvain Fleury, Nathalie Girard, Maëlys Le Magadou, Anthony Cherbonnier, Mickaël Renault, Eric Anquetil, and Eric Jamet. 2020. Can tablet apps support the learning of handwriting? an investigation of learning outcomes in kindergarten classroom. *Computers & Education*, 151:103831.
- Paul F Christiano, Jan Leike, Tom Brown, Miljan Matic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.
- Wei Dai, Jionghao Lin, Flora Jin, Tongguang Li, Yi-Shan Tsai, Dragan Gasevic, and Guanliang Chen. 2023. Can large language models provide feedback to students? a case study on chatgpt.
- Dorottya Demszky and Heather Hill. 2022. The NCTE Transcripts: A dataset of elementary math classroom transcripts. *arXiv preprint arXiv:2211.11772*.
- Dorottya Demszky and Jing Liu. 2023. M-Powering Teachers: Natural language processing powered feedback improves 1:1 instruction and student outcomes.
- Dorottya Demszky, Jing Liu, Heather Hill, Dan Jurafsky, and Chris Piech. 2023a. Can automated feedback improve teachers' uptake of student ideas? evidence from a randomized controlled trial in a large-scale online. *Education Evaluation and Policy Analysis (EEPA)*.
- Dorottya Demszky, Jing Liu, Heather C Hill, Dan Jurafsky, and Chris Piech. 2023b. Can automated feedback improve teachers' uptake of student ideas? evidence from a randomized controlled trial in a large-scale online course. *Educational Evaluation and Policy Analysis*.
- Laura M Desimone and Katie Pak. 2017. Instructional coaching as high-quality professional development. *Theory into practice*, 56(1):3–12.
- P. J. Donnelly, N. Blanchard, A. M. Olney, S. Kelly, M. Nystrand, and S. K. D'Mello. 2017. Words matter: Automatic detection of teacher questions in live classroom discourse using linguistics, acoustics and context. 218–227. Proceedings of the Seventh International Learning Analytics & Knowledge Conference on - LAK '17.
- Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. [Chatgpt outperforms crowd-workers for text-annotation tasks](#).
- Xingwei He, Zhenghao Lin, Yeyun Gong, A-Long Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, and Weizhu Chen. 2023. [Annollm: Making large language models to be better crowdsourced annotators](#).
- Heather C Hill, Merrie L Blunk, Charalambos Y Charalambous, Jennifer M Lewis, Geoffrey C Phelps, Laurie Sleep, and Deborah Loewenberg Ball. 2008. Mathematical knowledge for teaching and the mathematical quality of instruction: An exploratory study. *Cognition and instruction*, 26(4):430–511.
- Pihel Hunt, Äli Leijen, and Marieke van der Schaaf. 2021. Automated feedback is nice and human presence makes it better: Teachers' perceptions of feedback by means of an e-portfolio enhanced with learning analytics. *Education Sciences*, 11(6):278.
- Jennifer Jacobs, Karla Scornavacco, Charis Harty, Abhijit Suresh, Vivian Lai, and Tamara Sumner. 2022. Promoting rich discussions in mathematics classrooms: Using personalized, automated feedback to support reflection and instructional change. *Teaching and Teacher Education*, 112:103631.
- E. Jensen, M. Dale, P. J. Donnelly, C. Stone, S. Kelly, A. Godley, and S. K. D'Mello. 2020. Toward automated feedback on teacher discourse to enhance teacher learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. [Survey of hallucination in natural language generation](#). *ACM Computing Surveys*, 55(12):1–38.
- S. Kelly, A. M. Olney, P. Donnelly, M. Nystrand, and S. K. D'Mello. 2018. [Automatically measuring question authenticity in real-world classrooms](#). *Educational Researcher*, 47:7.

- Sean Kelly, Robert Bringe, Esteban Aucejo, and Jane Cooley Fruehwirth. 2020. Using global observation protocols to inform research on teaching effectiveness and school improvement: Strengths and emerging limitations. *Education Policy Analysis Archives*, 28:62–62.
- M. A. Kraft, D. Blazar, and D. Hogan. 2018. [The effect of teacher coaching on instruction and achievement: A meta-analysis of the causal evidence](#). *Review of Educational Research*, 88(4):547–588.
- Matthew A Kraft and David Blazar. 2018. Taking teacher coaching to scale: Can personalized training become standard practice? *Education Next*, 18(4):68–75.
- Taja Kuzman, Igor Mozetic, and Nikola Ljubešić. 2023. Chatgpt: Beginning of an end of manual linguistic data annotation? use case of automatic genre identification. *arXiv e-prints*, pages arXiv–2303.
- Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, et al. 2023. Summary of chatgpt/gpt-4 research and perspective towards the future of large language models. *arXiv preprint arXiv:2304.01852*.
- Felipe Martinez, Sandy Taut, and Kevin Schaaf. 2016. Classroom observation for evaluating and improving teaching: An international perspective. *Studies in Educational Evaluation*, 49:15–29.
- Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, et al. 2022. Teaching language models to support answers with verified quotes. *arXiv preprint arXiv:2203.11147*.
- Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. [Webgpt: Browser-assisted question-answering with human feedback](#).
- Zachary A. Pardos and Shreya Bhandari. 2023. [Learning gain differences between chatgpt and human tutor generated algebra hints](#).
- Robert C Pianta, Karen M La Paro, and Bridget K Hamre. 2008. *Classroom Assessment Scoring System™: Manual K-3*. Paul H Brookes Publishing.
- B. Samei, A. M. Olney, S. Kelly, M. Nystrand, S. D’Mello, N. Blanchard, X. Sun, M. Glaus, and A. Graesser. 2014. [Domain independent assessment of dialogic properties of classroom discourse](#).
- Baruch B Schwarz, Naomi Prusak, Osama Swidan, Adva Livny, Kobi Gal, and Avi Segal. 2018. Orchestrating the emergence of conceptual learning: A case study in a geometry class. *International Journal of Computer-Supported Collaborative Learning*, 13:189–211.
- Yen-Ning Su, Chia-Cheng Hsu, Hsin-Chin Chen, Kuo-Kuang Huang, and Yueh-Min Huang. 2014. Developing a sensor-based learning concentration detection system. *Engineering Computations*, 31(2):216–230.
- A. Suresh, J. Jacobs, V. Lai, C. Tan, W. Ward, J. H. Martin, and T. Sumner. 2021. [Using transformers to provide teachers with personalized feedback on their classroom discourse: The talkmoves application](#). arxiv. Preprint.
- Earl E Thomas, David L Bell, Maureen Spelman, and Jennifer Briody. 2015. The growth of instructional coaching partner conversations in a prek-3rd grade teacher professional development experience. *Journal of Adult Education*, 44(2):1–6.
- Walton Family Foundation. 2023. [ChatGPT Used by Teachers More Than Students, New Survey from Walton Family Foundation Finds](#).
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.
- Ted Wragg. 2011. *An introduction to classroom observation (Classic edition)*. Routledge.

### Example of Task A

Consider the following classroom transcript.

Transcript:

student: Because -  
teacher: Nice and loud so we can hear you.  
student: I did 12 times 3 equals 36.  
teacher: Okay, 12 times 3 equals 36.  
student: Times 2, 36 times 2.  
teacher: Then 36 times 2 equals 72. And that's your answer, right? Seventy-two inches. Where'd you get the 12 from? [...]

Based on the classroom transcript, rate the behavior management of the teacher on a scale of 1-7 (low-high). Behavior management encompasses the teacher's use of effective methods to encourage desirable behavior and prevent and re-direct misbehavior.

Rating (only specify a number between 1-7):

Model response

6

Figure 6: The top box shows an example of task A for directly predicting the scores (DA) for behavior management (CLBM). The bottom box shows the model's output. For space reasons, the full transcript has been cut out, indicated by [...].

## A Prompts and decoding parameters

This section provides all the prompts we used in our work and decoding parameters with using ChatGPT/gpt-3.5-turbo. We used the OpenAI API to send queries to ChatGPT. We sampled from the model with temperature 0.

The subsections include the prompts for (a) scoring the teacher according to the CLASS and MQI rubric, (b) identifying highlights and missed opportunities and (c) providing actionable insights for teachers.

### A.1 Observation scores

We prompt ChatGPT to provide scores according to the CLASS and MQI rubrics.

Prompts for directly predicting the scores are shown in:

- Figure 8 for CLPC.
- Figure 9 for CLBM
- Figure 10 for CLINSTD

- Figure 11 for EXPL
- Figure 12 for REMED
- Figure 13 for LANGIMP
- Figure 14 for SMQR

Prompts for directly predicting the scores with additional rubric descriptions are shown in:

- Figure 15 for CLPC.
- Figure 16 for CLBM
- Figure 17 for CLINSTD
- Figure 18 for EXPL
- Figure 19 for REMED
- Figure 20 for LANGIMP
- Figure 21 for SMQR

Prompts for reasoning then predicting the scores are shown in:

- Figure 22 for CLPC.
- Figure 23 for CLBM
- Figure 24 for CLINSTD
- Figure 25 for EXPL
- Figure 26 for REMED
- Figure 27 for LANGIMP
- Figure 28 for SMQR

### A.2 Highlights and missed opportunities

We prompt ChatGPT to identify highlights and missed opportunities according to the CLASS and MQI dimensions. The prompts for each dimension are shown in:

- Figure 29 for CLPC
- Figure 30 for CLBM
- Figure 31 for CLINSTD
- Figure 32 for EXPL
- Figure 33 for REMED
- Figure 34 for LANGIMP
- Figure 35 for SMQR

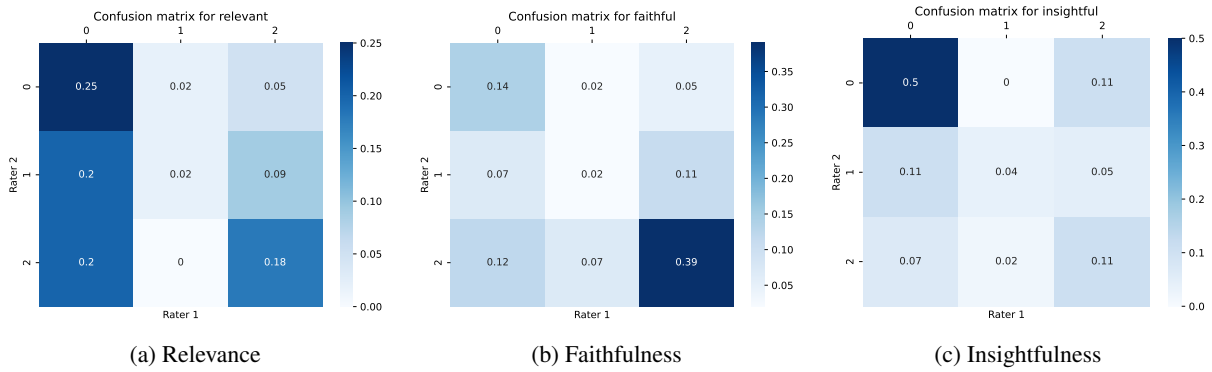


Figure 7: Confusion matrices between the two human raters on each of the criteria used in Task B: (a) *relevance*, (b) *faithfulness*, and (c) *insightfulness*.

### Prompt for direct score prediction (DA) on CLPC

Consider the following classroom transcript.

Transcript:  
{transcript}

Based on the classroom transcript, rate the positive climate of the classroom on a scale of 1-7 (low-high). Positive climate reflects the emotional connection and relationships among teachers and students, and the warmth, respect, and enjoyment communicated by verbal and non-verbal interactions.

Rating (only specify a number between 1-7):

Figure 8: Prompt for directly predicting the scores (DA) on the CLASS dimension CLPC.

### A.3 Actionable suggestions

We prompt ChatGPT to make actionable suggestions to the teacher for eliciting more student mathematical reasoning in the classroom. The prompt used for this task is shown in Figure 36.

## B Human experiments

We recruited 2 experienced human teachers to evaluate the generated model responses. As illustrated in our main figure (Figure 1), there are three main responses that are being evaluated by the human teachers: the highlights, missed opportunities and suggestions. Every observation code has their own generated highlights and missed opportunities.

### B.1 Collecting model responses to evaluate

**Highlights and missed opportunities** From the transcripts which have less than 10% student contributions including “[inaudible]” markers, we sample 18 random 15-minutes transcript segments for the CLASS codes, and 18 random 7.5 minutes tran-

script segments for the MQI codes. Every code has 2 model-generated highlights and missed opportunities. In total, we have 216 **CLASS-annotated items**. The calculation is: 18 segments  $\times$  3 CLASS codes  $\times$  (2 highlights + 2 missed opportunities) = 216 items. In total, we have 288 **MQI-annotated items**. The calculation is: 18 segments  $\times$  4 MQI codes  $\times$  (2 highlights + 2 missed opportunities) = 288 items.

**Suggestions** We use the same 18 random MQI 7.5-minutes transcript segments for prompting the model for suggestions. In total, we have 36 **item suggestions**. The calculation is 18 segments  $\times$  2 suggestions = 36 items.

### B.2 Evaluation axes and human interface

This section details what we ask the teachers to evaluate qualitatively. Some of the details are repeated from Section 4.2 for completeness. We additionally include screenshots of the human experiment interface.

### Prompt for direct score prediction (DA) on CLBM

Consider the following classroom transcript.

Transcript:  
{transcript}

Based on the classroom transcript, rate the behavior management of the teacher on a scale of 1-7 (low-high). Behavior management encompasses the teacher's use of effective methods to encourage desirable behavior and prevent and re-direct misbehavior.

Rating (only specify a number between 1-7):

Figure 9: Prompt for directly predicting the scores (DA) on the CLASS dimension CLBM.

### Prompt for direct score prediction (DA) on CLINSTD

Consider the following classroom transcript.

Transcript:  
{transcript}

Based on the classroom transcript, rate the instructional dialogue of the teacher on a scale of 1-7 (low-high). Instructional dialogue captures the purposeful use of content-focused discussion among teachers and students that is cumulative, with the teacher supporting students to chain ideas together in ways that lead to deeper understanding of content. Students take an active role in these dialogues and both the teacher and students use strategies that facilitate extended dialogue.

Rating (only specify a number between 1-7):

Figure 10: Prompt for directly predicting the scores (DA) on the CLASS dimension CLINSTD.

**Highlights and missed opportunities** The teachers evaluate the model examples along three axes. One is **relevance**: Is the model's response relevant to the CLASS or MQI dimension of interest? Two is **faithfulness**: Does the model's response have the right interpretation of the events that occur in the classroom transcript? We evaluate along this dimension because the model sometimes can hallucinate or misinterpret the events in the transcript when providing examples. Three is **insightfulness**: Does the model's response reveal something beyond the line segment's obvious meaning in the transcript? We ask the teachers to evaluate on a 3-point scale (yes, somewhat, no). Optionally, the teacher may additionally provide a free text comment, if they want to elaborate their answer.

Figure 37 shows the human interface for evaluating the CLASS observation items, and Figure 38 for evaluating the MQI observation items.

**Suggestions** The teachers evaluate the model suggestions along four axes. One is **relevance**: Is the model's response relevant to eliciting more student mathematical reasoning in the classroom? Two is **faithfulness**: Does the model's response have the right interpretation of the events that occur in the classroom transcript? Similar to the previous research question, we evaluate along this dimension because the model sometimes can hallucinate or misinterpret the events in the transcript when providing suggestions. Three is **actionability**: Is the model's suggestion something that the teacher can easily translate into practice for improving their

### Prompt for direct score prediction (DA) on EXPL

Consider the following classroom transcript.

Transcript:  
{transcript}

Based on the classroom transcript, rate the teacher's mathematical explanations on a scale of 1-3 (low-high). Mathematical explanations focus on the why, eg. why a procedure works, why a solution method is (in)appropriate, why an answer is true or not true, etc. Do not count 'how', eg. description of the steps, or definitions unless meaning is also attached.

Rating (only specify a number between 1-3):

Figure 11: Prompt for directly predicting the scores (DA) on the MQI dimension EXPL.

### Prompt for direct score prediction (DA) on REMED

Consider the following classroom transcript.

Transcript:  
{transcript}

Based on the classroom transcript, rate the teacher's degree of remediation of student errors and difficulties on a scale of 1-3 (low-high). This means that the teacher gets at the root of student misunderstanding, rather than repairing just the procedure or fact. This is more than a simple correction of a student mistake.

Rating (only specify a number between 1-3):

Figure 12: Prompt for directly predicting the scores (DA) on the MQI dimension REMED.

teaching or encouraging student mathematical reasoning? Finally, four is **novelty**: Is the model suggestion something that the teacher already does in the transcript? Similar to the previous section, we ask the teachers to evaluate on a 3-point scale (yes, somewhat, no).

Figure 39 shows the human interface for evaluating the model suggestions.

## C Additional results on quantitative scoring

We include the additional results on the the quantitative scoring task.

**CLASS** Figure 40 shows scatter plots of the model predicted scores vs. the human scores. It shows this across CLASS observation items and

prompting methods (DA, DA<sup>+</sup>, and RA). Figure 41 shows the same data, but compares the human and model predicted score distribution.

**MQI** Figure 42 shows scatter plots of the model predicted scores vs. the human scores. It shows this across MQI observation items and prompting methods (DA, DA<sup>+</sup>, and RA). Figure 43 shows the same data, but compares the human and model predicted score distribution.

### C.1 Interrater Agreement

We compute interrater agreement on the examples that both teachers rated (20%). Since our goal was to collect teachers' unbiased perceptions, we did not conduct any calibration for this task; we leave this for future work. For task B, we measure a Cohen's kappa with linear weighting of 0.16 for *rele-*



### Prompt for direct score prediction (DA) on LANGIMP

Consider the following classroom transcript.

Transcript:

{transcript}

Based on the classroom transcript, rate the teacher's imprecision in language or notation on a scale of 1-3 (low-high). The teacher's imprecision in language or notation refers to problematic uses of mathematical language or notation. For example, errors in notation (eg. mathematical symbols), in mathematical language (eg. technical mathematical terms like "equation") or general language (eg. explaining mathematical ideas or procedures in non-technical terms). Do not count errors that are noticed and corrected within the segment.

Rating (only specify a number between 1-3):

Figure 13: Prompt for directly predicting the scores (DA) on the MQI dimension LANGIMP.

vance, 0.23 for *faithfulness*, and 0.32 for *insightfulness*. Figure 7 illustrates why there is particularly low agreement on relevance: One rater tends to select more extreme values for relevance, whereas the other rater selects more uniformly across the values. This results in low agreement for relevance. The Cohen's kappas with quadratic weighting are 0.23 for *relevance*, 0.36 for *faithfulness*, and 0.37 for *insightfulness*. The Cohen's kappas with quadratic weighting is slightly higher as it adjusts the penalty between scores 1 and 3 to be different from the penalty between scores 1 and 2 for instance. For Task C, we only have 2 examples per criterion, which is too sparse for computing Cohen's kappa.

### D Examples of Transcripts, Model Responses, and Human Evaluations

Figure 44 shows a concrete example of the suggestions prompt given to the model. Figure ?? then shows one of the suggestions that the model generates. Figure 45 then shows the ratings provided from one of the human annotators on that suggestion.

**Prompt for direct score prediction (DA) on SMQR**

Consider the following classroom transcript.

Transcript:  
{transcript}

Based on the classroom transcript, rate the degree of student mathematical questioning and reasoning on a scale of 1-3 (low-high). Student mathematical questioning and reasoning means that students engage in mathematical thinking. Examples include but are not limited to: Students provide counter-claims in response to a proposed mathematical statement or idea, ask mathematically motivated questions requesting explanations, make conjectures about the mathematics discussed in the lesson, etc.

Rating (only specify a number between 1-3):

Figure 14: Prompt for directly predicting the scores (DA) on the MQI dimension SMQR.

**Prompt with rubric description for direct score prediction (DA<sup>+</sup>) on CLPC**

Consider the following classroom transcript.

Transcript:  
{transcript}

Based on the classroom transcript, rate the positive climate of the classroom on a scale of 1-7 (low-high). Positive climate reflects the emotional connection and relationships among teachers and students, and the warmth, respect, and enjoyment communicated by verbal and non-verbal interactions.

Explanation of ratings:

1, 2: The teacher and students seem distant from one another, display flat affect, do not provide positive comments, or rarely demonstrate respect for one another.

3, 4, 5: There is some display of a supportive relationship, of positive affect, of positive communication, or of respect between the teacher and the students.

6, 7: There are many displays of a supportive relationship, of positive affect, of positive communication, or of respect between the teacher and the students.

Rating (only specify a number between 1-7):

Figure 15: Prompt for directly predicting the scores (DA<sup>+</sup>) on the CLASS dimension CLPC.

**Prompt with rubric description for direct score prediction (DA<sup>+</sup>) on CLBM**

Consider the following classroom transcript.

Transcript:  
{transcript}

Based on the classroom transcript, rate the behavior management of the teacher on a scale of 1-7 (low-high). Behavior management encompasses the teacher's use of effective methods to encourage desirable behavior and prevent and re-direct misbehavior.

Explanation of ratings:

1, 2: Teacher does not set expectations of the rules or inconsistently enforces them, teacher is reactive to behavioral issues or does not monitor students, teacher uses ineffective methods to redirect misbehavior, students are defiant.

3, 4, 5: Teacher sets some expectations of the rules but inconsistently enforces them, teacher uses a mix of proactive and reactive approaches to behavioral issues and sometimes monitors students, teacher uses a mix of effective and ineffective strategies to misdirect behavior, students periodically misbehave.

6, 7: Teacher sets clear expectations of the rules, teacher is proactive and monitors students, teacher consistently uses effective strategies to redirect mishavior, students are compliant.

Rating (only specify a number between 1-7):

Figure 16: Prompt for directly predicting the scores (DA<sup>+</sup>) on the CLASS dimension CLBM.

### **Prompt with rubric description for direct score prediction (DA<sup>+</sup>) on CLINSTD**

Consider the following classroom transcript.

Transcript:  
{transcript}

Based on the classroom transcript, rate the instructional dialogue of the teacher on a scale of 1-7 (low-high). Instructional dialogue captures the purposeful use of content-focused discussion among teachers and students that is cumulative, with the teacher supporting students to chain ideas together in ways that lead to deeper understanding of content. Students take an active role in these dialogues and both the teacher and students use strategies that facilitate extended dialogue.

Explanation of ratings:

1, 2: There are no or few discussions in class or discussions unrelated to content, class is dominated by teacher talk, the teacher and students ask closed questions or rarely acknowledge/repeat/extend others' comments.

3, 4, 5: There are occasional brief content-based discussions in class among teachers and students, the class is mostly dominated by teacher talk, the teacher and students sometimes use facilitation strategies to encourage more elaborated dialogue.

6, 7: There are frequent, content-driven discussions in the class between teachers and students, class dialogues are distributed amongst the teacher and the majority of students, the teacher and students frequently use facilitation strategies that encourage more elaborated dialogue.

Rating (only specify a number between 1-7):

Figure 17: Prompt for directly predicting the scores (DA<sup>+</sup>) on the CLASS dimension CLINSTD.

**Prompt with rubric description for direct score prediction (DA<sup>+</sup>) on EXPL**

Consider the following classroom transcript.

Transcript:

{transcript}

Based on the classroom transcript, rate the teacher's mathematical explanations on a scale of 1-3 (low-high). Mathematical explanations focus on the why, eg. why a procedure works, why a solution method is (in)appropriate, why an answer is true or not true, etc. Do not count 'how', eg. description of the steps, or definitions unless meaning is also attached.

Explanation of ratings:

- 1: A mathematical explanation occurs as an isolated instance in the segment.
- 2: Two or more brief explanations occur in the segment OR an explanation is more than briefly present but not the focus of instruction.
- 3: One of more mathematical explanation(s) is a focus of instruction in the segment. The explanation(s) need not be most or even a majority of the segment; what distinguishes a High is the fact that the explanation(s) are a major feature of the teacher-student work (e.g., working for 2-3 minutes to elucidate the simplifying example above).

Rating (only specify a number between 1-3):

Figure 18: Prompt for directly predicting the scores (DA<sup>+</sup>) on the CLASS dimension EXPL.

**Prompt with rubric description for direct score prediction (DA<sup>+</sup>) on REMED**

Consider the following classroom transcript.

Transcript:

{transcript}

Based on the classroom transcript, rate the teacher's degree of remediation of student errors and difficulties on a scale of 1-3 (low-high). This means that the teacher gets at the root of student misunderstanding, rather than repairing just the procedure or fact. This is more than a simple correction of a student mistake.

Explanation of ratings:

- 1: Brief conceptual or procedural remediation occurs.
- 2: Moderate conceptual or procedural remediation occurs or brief pre-remediation (calling students' attention to a common error) occurs.
- 3: Teach engages in conceptual remediation systematically and at length. Examples include identifying the source of student errors or misconceptions, discussing how student errors illustrate broader misunderstanding and then addressing those errors, or extended pre-remediation.

Rating (only specify a number between 1-3):

Figure 19: Prompt for directly predicting the scores (DA<sup>+</sup>) on the CLASS dimension REMED.

**Prompt with rubric description for direct score prediction (DA<sup>+</sup>) on LANGIMP**

Consider the following classroom transcript.

Transcript:

{transcript}

Based on the classroom transcript, rate the teacher's imprecision in language or notation on a scale of 1-3 (low-high). The teacher's imprecision in language or notation refers to problematic uses of mathematical language or notation. For example, errors in notation (eg. mathematical symbols), in mathematical language (eg. technical mathematical terms like "equation") or general language (eg. explaining mathematical ideas or procedures in non-technical terms). Do not count errors that are noticed and corrected within the segment.

Explanation of ratings:

- 1: Brief instance of imprecision. Does not obscure the mathematics of the segment.
- 2: Imprecision occurs in part(s) of the segment or imprecision obscures the mathematics but for only part of the segment.
- 3: Imprecision occurs in most or all of the segment or imprecision obscures the mathematics of the segment.

Rating (only specify a number between 1-3):

Figure 20: Prompt for directly predicting the scores (DA<sup>+</sup>) on the CLASS dimension LANGIMP.

### Prompt with rubric description for direct score prediction (DA<sup>+</sup>) on SMQR

Consider the following classroom transcript.

Transcript:  
{transcript}

Based on the classroom transcript, rate the degree of student mathematical questioning and reasoning on a scale of 1-3 (low-high). Student mathematical questioning and reasoning means that students engage in mathematical thinking. Examples include but are not limited to: Students provide counter-claims in response to a proposed mathematical statement or idea, ask mathematically motivated questions requesting explanations, make conjectures about the mathematics discussed in the lesson, etc.

Explanation of ratings:

- 1: One of two instances of brief student mathematical questioning or reasoning are present.
- 2: Student mathematical questioning or reasoning is more sustained or more frequent, but it is not characteristic of the segment.
- 3: Student mathematical questioning or reasoning characterizes much of the segment.

Rating (only specify a number between 1-3):

Figure 21: Prompt for directly predicting the scores (DA<sup>+</sup>) on the CLASS dimension SMQR.

### Prompting with reasoning, then predicting the score (RA) on CLPC

Consider the following classroom transcript.

Transcript:  
{transcript}

Please do the following.

1. Think step-by-step how you would rate the positive climate of the classroom on a scale of 1-7 (low-high). Positive climate reflects the emotional connection and relationships among teachers and students, and the warmth, respect, and enjoyment communicated by verbal and non-verbal interactions.
2. Provide your rating as a number between 1 and 7.

Format your answer as:

Reasoning:

Rating (only specify a number between 1-7):

Reasoning:

Figure 22: Prompt for reasoning, then predicting the score (RA) on the CLASS dimension CLPC.



### Prompting with reasoning, then predicting the score (RA) on CLBM

Consider the following classroom transcript.

Transcript:  
{transcript}

Please do the following.

1. Think step-by-step how you would rate the behavior management of the teacher on a scale of 1-7 (low-high). Behavior management encompasses the teacher's use of effective methods to encourage desirable behavior and prevent and re-direct misbehavior.
2. Provide your rating as a number between 1 and 7.

Format your answer as:

Reasoning:

Rating (only specify a number between 1-7):

Reasoning:

Figure 23: Prompt for reasoning, then predicting the score (RA) on the CLASS dimension CLBM.

### Prompting with reasoning, then predicting the score (RA) on CLINSTD

Consider the following classroom transcript.

Transcript:  
{transcript}

Please do the following.

1. Think step-by-step how you would rate the instructional dialogue of the teacher on a scale of 1-7 (low-high). Instructional dialogue captures the purposeful use of content-focused discussion among teachers and students that is cumulative, with the teacher supporting students to chain ideas together in ways that lead to deeper understanding of content. Students take an active role in these dialogues and both the teacher and students use strategies that facilitate extended dialogue.
2. Provide your rating as a number between 1 and 7.

Format your answer as:

Reasoning:

Rating (only specify a number between 1-7):

Reasoning:

Figure 24: Prompt for reasoning, then predicting the score (RA) on the CLASS dimension CLINSTD.

**Prompting with reasoning, then predicting the score (RA) on EXPL**

Consider the following classroom transcript.

Transcript:

{transcript}

Please do the following.

1. Think step-by-step how you would rate the teacher's mathematical explanations on a scale of 1-3 (low-high). Mathematical explanations focus on the why, eg. why a procedure works, why a solution method is (in)appropriate, why an answer is true or not true, etc. Do not count 'how', eg. description of the steps, or definitions unless meaning is also attached.
2. Provide your rating as a number between 1 and 3.

Format your answer as:

Reasoning:

Rating (only specify a number between 1-3):

Reasoning:

Figure 25: Prompt for reasoning, then predicting the score (RA) on the CLASS dimension EXPL.

**Prompting with reasoning, then predicting the score (RA) on REMED**

Consider the following classroom transcript.

Transcript:

{transcript}

Please do the following.

1. Think step-by-step how you would rate the teacher's degree of remediation of student errors and difficulties on a scale of 1-3 (low-high). This means that the teacher gets at the root of student misunderstanding, rather than repairing just the procedure or fact. This is more than a simple correction of a student mistake.
2. Provide your rating as a number between 1 and 3.

Format your answer as:

Reasoning:

Rating (only specify a number between 1-3):

Reasoning:

Figure 26: Prompt for reasoning, then predicting the score (RA) on the CLASS dimension REMED.

### Prompting with reasoning, then predicting the score (RA) on LANGIMP

Consider the following classroom transcript.

Transcript:  
{transcript}

Please do the following.

1. Think step-by-step how you would rate the teacher's imprecision in language or notation on a scale of 1-3 (low-high). The teacher's imprecision in language or notation refers to problematic uses of mathematical language or notation. For example, errors in notation (eg. mathematical symbols), in mathematical language (eg. technical mathematical terms like "equation") or general language (eg. explaining mathematical ideas or procedures in non-technical terms). Do not count errors that are noticed and corrected within the segment.
2. Provide your rating as a number between 1 and 3.

Format your answer as:

Reasoning:

Rating (only specify a number between 1-3):

Reasoning:

Figure 27: Prompt for reasoning, then predicting the score (RA) on the CLASS dimension LANGIMP.

### Prompting with reasoning, then predicting the score (RA) on SMQR

Consider the following classroom transcript.

Transcript:  
{transcript}

Please do the following.

1. Think step-by-step how you would rate the degree of student mathematical questioning and reasoning on a scale of 1-3 (low-high). Student mathematical questioning and reasoning means that students engage in mathematical thinking. Examples include but are not limited to: Students provide counter-claims in response to a proposed mathematical statement or idea, ask mathematically motivated questions requesting explanations, make conjectures about the mathematics discussed in the lesson, etc.
2. Provide your rating as a number between 1 and 3.

Format your answer as:

Reasoning:

Rating (only specify a number between 1-3):

Reasoning:

Figure 28: Prompt for reasoning, then predicting the score (RA) on the CLASS dimension SMQR.

### Prompt for identifying highlights and missed opportunity on CLPC

Consider the following classroom transcript.

Transcript:  
{transcript}

Please do the following.

1. Provide up to 5 good examples of the classroom's positive climate. Positive climate reflects the emotional connection and relationships among teachers and students, and the warmth, respect, and enjoyment communicated by verbal and non-verbal interactions.
2. Provide up to 5 bad examples (eg. missed opportunities or poor execution) of the classroom's positive climate.

Format your answer as:

Good examples

1. Line number: <specify line number>, Segment: "<copied from transcript>", Reason: <specify why this is a good example>
2. ...

Bad examples

1. Line number: <specify line number>, Segment: "<copied from transcript>", Reason: <specify why this is a bad example>
2. ...

Good examples:

Figure 29: Prompt for identifying highlights and missed opportunity on CLPC.

### Prompt for identifying highlights and missed opportunity on CLBM

Consider the following classroom transcript.

Transcript:  
{transcript}

Please do the following.

1. Provide up to 5 good examples of the teacher's behavior management. Behavior management encompasses the teacher's use of effective methods to encourage desirable behavior and prevent and re-direct misbehavior.
2. Provide up to 5 bad examples (eg. missed opportunities or poor execution) of the teacher's behavior management.

Format your answer as:

Good examples

1. Line number: <specify line number>, Segment: "<copied from transcript>", Reason: <specify why this is a good example>
2. ...

Bad examples

1. Line number: <specify line number>, Segment: "<copied from transcript>", Reason: <specify why this is a bad example>
2. ...

Good examples:

Figure 30: Prompt for identifying highlights and missed opportunity on CLBM.

### Prompt for identifying highlights and missed opportunity on CLINSTD

Consider the following classroom transcript.

Transcript:  
{transcript}

Please do the following.

1. Provide up to 5 good examples of the teacher's instructional dialogue. Instructional dialogue captures the purposeful use of content-focused discussion among teachers and students that is cumulative, with the teacher supporting students to chain ideas together in ways that lead to deeper understanding of content. Students take an active role in these dialogues and both the teacher and students use strategies that facilitate extended dialogue.
2. Provide up to 5 bad examples of (eg. missed opportunities or poor execution) the teacher's instructional dialogue.

Format your answer as:

Good examples

1. Line number: <specify line number>, Segment: "<copied from transcript>", Reason: <specify why this is a good example>
2. ...

Bad examples

1. Line number: <specify line number>, Segment: "<copied from transcript>", Reason: <specify why this is a bad example>
2. ...

Good examples:

Figure 31: Prompt for identifying highlights and missed opportunity on CLINSTD.

### Prompt for identifying highlights and missed opportunity on EXPL

Consider the following classroom transcript.

Transcript:  
{transcript}

Please do the following.

1. Provide up to 5 good examples of the teacher's mathematical explanations. Mathematical explanations focus on the why, eg. why a procedure works, why a solution method is (in)appropriate, why an answer is true or not true, etc. Do not count 'how', eg. description of the steps, or definitions unless meaning is also attached.
2. Provide up to 5 bad examples (eg. missed opportunities or poor execution) of the teacher's mathematical explanations.

Format your answer as:

Good examples

1. Line number: <specify line number>, Segment: "<copied from transcript>", Reason: <specify why this is a good example>
2. ...

Bad examples

1. Line number: <specify line number>, Segment: "<copied from transcript>", Reason: <specify why this is a bad example>
2. ...

Good examples:

Figure 32: Prompt for identifying highlights and missed opportunity on EXPL.

### Prompt for identifying highlights and missed opportunity on REMED

Consider the following classroom transcript.

Transcript:  
{transcript}

Please do the following.

1. Provide up to 5 good examples of the teacher's remediation of student errors and difficulties. This means that the teacher gets at the root of student misunderstanding, rather than repairing just the procedure or fact. This is more than a simple correction of a student mistake.
2. Provide up to 5 bad examples (eg. missed opportunities or poor execution) of the teacher's remediation of student errors and difficulties.

Format your answer as:

Good examples

1. Line number: <specify line number>, Segment: "<copied from transcript>", Reason: <specify why this is a good example>
2. ...

Bad examples

1. Line number: <specify line number>, Segment: "<copied from transcript>", Reason: <specify why this is a bad example>
2. ...

Good examples:

Figure 33: Prompt for identifying highlights and missed opportunity on REMED.



### Prompt for identifying highlights and missed opportunity on LANGIMP

Consider the following classroom transcript.

Transcript:  
{transcript}

Please do the following.

1. Provide up to 5 good examples of the teacher's imprecision in language or notation. The teacher's imprecision in language or notation refers to problematic uses of mathematical language or notation. For example, errors in notation (eg. mathematical symbols), in mathematical language (eg. technical mathematical terms like "equation") or general language (eg. explaining mathematical ideas or procedures in non-technical terms). Do not count errors that are noticed and corrected within the segment.
2. Provide up to 5 bad examples (eg. missed opportunities or poor execution) of the teacher's imprecision in language or notation.

Format your answer as:

Good examples

1. Line number: <specify line number>, Segment: "<copied from transcript>", Reason: <specify why this is a good example>
2. ...

Bad examples

1. Line number: <specify line number>, Segment: "<copied from transcript>", Reason: <specify why this is a bad example>
2. ...

Good examples:

Figure 34: Prompt for identifying highlights and missed opportunity on LANGIMP.

### Prompt for identifying highlights and missed opportunity on SMQR

Consider the following classroom transcript.

Transcript:  
{transcript}

Please do the following.

1. Provide up to 5 good examples of the students' mathematical questioning and reasoning. Student mathematical questioning and reasoning means that students engage in mathematical thinking. Examples include but are not limited to: Students provide counter-claims in response to a proposed mathematical statement or idea, ask mathematically motivated questions requesting explanations, make conjectures about the mathematics discussed in the lesson, etc.
2. Provide up to 5 bad examples (eg. missed opportunities or poor execution) of the students' mathematical questioning and reasoning.

Format your answer as:

Good examples

1. Line number: <specify line number>, Segment: "<copied from transcript>", Reason: <specify why this is a good example>
2. ...

Bad examples

1. Line number: <specify line number>, Segment: "<copied from transcript>", Reason: <specify why this is a bad example>
2. ...

Good examples:

Figure 35: Prompt for identifying highlights and missed opportunity on SMQR.

### **Prompt for suggestions on eliciting more student reasoning in the classroom**

Consider the following classroom transcript.

Transcript:  
{transcript}

The transcript contains many short student responses. Please provide 5 suggestions for the teacher on how the teacher could elicit more student reasoning in the classroom. Student reasoning is counted broadly as students asking questions, engaging in mathematical discourse with their teacher or peers, and providing explanations such as justifying their answers.

Format your answer as:

Advice to the teacher:

1. Line number: <specify line number>, Segment: "<copied from transcript>", Suggestion: <specify advice to the teacher>
2. ...

Advice to the teacher:

Figure 36: Prompt for suggestions on eliciting more student mathematical reasoning in the classroom.

## Evaluating Model Examples

### Instructions

**Setup:** You will be given a snippet of a classroom transcript and feedback from the model. The model feedback contains 2 examples of what the model thinks is good (eg. good execution) and bad (eg. missed opportunities) about certain aspects of the classroom transcript.

**Task:** You will be asked to evaluate the quality of the model examples along certain dimensions like whether the examples mentioned are actually present in the transcript or whether the feedback is useful for the teacher.

Current progress: 0 % completed, 0 / 120

### Transcript #2376, Good Example #1 on the teacher's behavior management

1. student: Because --  
2. teacher: Nice and loud so we can hear you.  
3. student: I did 12 times 3 equals 36.  
4. teacher: Okay, 12 times 3 equals 36.  
5. student: Times 2, 36 times 2.  
6. teacher: Then 36 times 2 equals 72. And that's your answer, right? Seventy-two inches. Where'd you get the 12 from?  
7. student: Well, 12 inches equals one foot so 12 inches.  
8. teacher: All right, so 12 inches equals one foot. Twelve inches equal one foot. So you knew that 12 times 3 is 36. Why did you do 12 times 3?  
9. student: Cause that equals one yard.  
10. teacher: Say that again?  
11. student: Three feet equals one yard.  
12. teacher: Okay, you know that three feet equals one yard. So how many yards -- how many -- it didn't ask you this, but how many inches --  
13. student: Are in a yard?  
14. teacher: Yeah.  
15. student: It's 34.  
16. teacher: How many inches are in one yard? It didn't ask you this. Student A, why don't you trade them? Student T, how many inches are in one yard?  
17. student: Eighteen?  
18. teacher: Eighteen? We have 18. What do you say, Student J?  
19. student: Twenty-four.  
20. teacher: Twenty-four.

#### Model prompt

Provide a good example of the teacher's behavior management. Behavior management encompasses the teacher's use of effective methods to encourage desirable behavior and prevent and re-direct misbehavior.

#### Model response #1

Line number: 2, Segment: "Nice and loud so we can hear you.", Reason: The teacher encourages the student to speak up and be heard.

Please evaluate the model response along the following 3 dimensions:

#### Rate how relevant of the model's example to the prompt

**Definition of relevance:** The model returns a response that is related to the prompt shown in the blue box.

**Examples:** An example of a relevant model response is: If the prompt asks to provide a good example of the teacher's classroom management, then the model's response pulls out a segment showing the teacher's classroom management, eg. "Segment: Everyone, let's use our indoor voices." An example of an irrelevant model response is "Segment: I think the answer is 5" (this is not related to the prompt).

- Not relevant  
 Somewhat relevant  
 Relevant

(Optional) Comments: eg. why is this relevant or not relevant?

#### Rate how faithful of an interpretation the model response is

**Definition of faithfulness:** The model response has the right interpretation of the events that occur in the classroom transcript. We evaluate along this dimension because the model sometimes can hallucinate or misinterpret the events in the transcript when providing suggestions.

**Examples:** An example of a faithful model response is "Line number: 22, Segment: "Could you repeat what you said?"; Reason: The teacher is asking the student to repeat what he said". An example of an unfaithful model response is "Reason: The teacher uses an aggressive tone to force the students to answer" even though the teacher is not threatening the students and no information about tone is in the transcript.

- Not faithful  
 Somewhat faithful  
 Faithful

(Optional) Comments: eg. why is this faithful or not faithful?

#### Rate how insightful the model response is

**Definition of insightfulness:** The model response is insightful if its reason points to something that's not obvious when only reading that single line of the transcript, and might require some knowledge about classroom dynamics.

**Examples:** An example of an insightful model response is "Line number: 2, Segment: "Okay, hold that thought..."; Reason: The teacher recognizes that many students have a similar question, and she puts the current activity on hold to address it. This then leads to a productive classroom discussion." This response is insightful, because it connects the teacher's actions to future implications for the class. An example of an un insightful model response is "Line number: 25, Segment: "Five feet wide equals 40 square what?"; Reason: The teacher is emphasizing the importance of units in measuring area and prompting the student to include the unit "feet" in their answer." This response is not insightful, because the interpretation is obvious from just this line of the transcript.

- Not insightful  
 Somewhat insightful  
 Insightful

(Optional) Comments: eg. why is this insightful or not insightful?

Note: The "Continue" button will be disabled until you've indicated your ratings on the 3 dimensions.

CONTINUE

Figure 37: Human interface for evaluating the highlights (good examples) and missed opportunities (bad examples) on CLASS observation items generated by the model.

## Evaluating Model Examples

### Instructions

**Setup:** You will be given a snippet of a classroom transcript and feedback from the model. The model feedback contains 2 examples of what the model thinks is good (eg. good execution) and bad (eg. missed opportunities) about certain aspects of the classroom transcript.

**Task:** You will be asked to evaluate the quality of the model examples along certain dimensions like whether the examples mentioned are actually present in the transcript or whether the feedback is useful for the teacher.

Current progress: 0% completed, 0 / 160

### Transcript #2776, Good Example #1 on the teachers's mathematical explanation

1. student: Divisor.  
2. teacher: Divided. It means the numerator divided by the denominator. That's what that line means. We also know that when we see, when something's in division it also looks like this. That means divided by, but here it's just the line. Then we see numbers right across the top here. Let's read these numbers with me everybody. Can everybody see? What are they? 12.  
3. teacher: Good. Then along the sides we see some more numbers. We've been forming one, two, three, four, five, six, seven, eight, nine, 10, 11, 12. So the numbers across the top are called the what? Numerators. The numbers down the side are called what? Denominators. So this is almost going to be like a division table. It's similar to a multiplication table. So the numbers that we're gonna divide are shown across the top and on the left sides and then we've gotta put the answers in all these little boxes. So we're going to use this division table to record decimal equivalents. What does that mean? If they're decimal equivalents of the fractions, what does that mean? Are they gonna be equal to the fractions? Numbers represent the numerators up here beside the denominators. So let's take a look at see what we're gonna do here. Today we're gonna write the decimal equivalent in those little boxes. Now what decimal would be equal to 0.5? Do you see that on there? What fraction is going to be equivalent to the decimal? Which one is that going to be, Student A?  
4. student: 2/4?  
5. teacher: 2/4? I'm going to tell you to look again. Yeah, 2/4 is actually equal to a half, but in this case I'm gonna put something that's already in the box. One divided by 2 is going to be 0.5. So what does 0.5 – what is it equivalent to, what fraction?  
6. student: One half.  
7. teacher: So I want you to write that in your box. So if we said one divided by two we'd put 0.5. Student A, you're one step ahead of us but if we look at the two up here, if I say two divided by four, what would that be everybody? Two divided by four. Here's the two divided by the four, what's my answer? 0.5. So you ought to put it in the box also. That's also equal to a half, right? So we know 2/4 equals a half. We know 1/2 is equal to the 0.5. These are the decimal equivalents. So we've got two of them in there. Let's look at three in the numerator divided by the six. And you can do that on your calculator. All of you know how to use the calculator right? Is there anybody who doesn't? So if I said to you let us look at 3 numerator divided by 6. So I'm gonna look at my calculator and I'm gonna press 3 then the divided by sign, 6 equals – what's that gonna equal?  
8. teacher: Where am I gonna put that one? Three divided by 6. Am I gonna put it right here?  
9. teacher: So you can do that. Three divided by 6 is going to be again, 0.5. So again we know 3/6 equals a half and it's equal to 25 or 0.5. So put it on your table. Put it on your table. Now do you see another one that we can take the numerator and divide it by the denominator and again come out with 0.5 – who can give me another one? Student N?  
10. student: Four divided by 8.  
11. teacher: So I look at my four as the numerator, I look at my 8 and I'm coming right over here. What am I gonna put in there? 0.5. So I'm putting here's my 4 and then I move my finger over to the 8 and again 4/8 is equal to a half so we'll put 0.5 here too. Do I see a pattern going on here?

### Model prompt

Provide a good example of the teacher's mathematical explanations. Mathematical explanations focus on the why, eg. why a procedure works, why a solution method is (in)appropriate, why an answer is true or not true, etc. Do not count 'how', eg. description of the steps, or definitions unless meaning is also attached.

### Model response #1

Line number: 9, Segment: "2/4 is actually equal to a half, but in this case I'm gonna put something that's already in the box. One divided by 2 is going to be 0.5.", Reason: The teacher explains why a different decimal equivalent is being used in this case, which helps students understand the concept better.

Please evaluate the model response along the following 3 dimensions:

### Rate how relevant of the model's example to the prompt

**Definition of relevance:** The model returns a response that is related to the prompt shown in the blue box.

**Examples:** An example of a relevant model response is: "If the prompt asks to provide a good example of the teacher's classroom management, then the model's response pulls out a segment showing the teacher's classroom management, eg. "Segment: Everyone, let's use our indoor voices." An example of an irrelevant model response is "Segment: I think the answer is 5" (this is not related to the prompt).

- Not relevant  
 Somewhat relevant  
 Relevant

(Optional) Comments: eg. why is this relevant or not relevant?

### Rate how faithful of an interpretation the model response is

**Definition of faithfulness:** The model response has the right interpretation of the events that occur in the classroom transcript. We evaluate along this dimension because the model sometimes can hallucinate or misinterpret the events in the transcript when providing suggestions.

**Examples:** An example of a faithful model response is "Line number: 22, Segment: "Could you repeat what you said?"; Reason: The teacher is asking the student to repeat what he said". An example of an unfaithful model response is "Reason: The teacher uses an aggressive tone to force the students to answer" even though the teacher is not threatening the students and no information about tone is in the transcript.

- Not faithful  
 Somewhat faithful  
 Faithful

(Optional) Comments: eg. why is this faithful or not faithful?

### Rate how insightful the model response is

**Definition of insightfulness:** The model response is insightful if its reason points to something that's not obvious when only reading that single line of the transcript, and might require some knowledge about classroom dynamics.

**Examples:** An example of an insightful model response is "Line number: 2, Segment: "Okay, hold that thought..."; Reason: The teacher recognizes that many students have a similar question, and she puts the current activity on hold to address it. This then leads to a productive classroom discussion." This response is insightful, because it connects the teacher's actions to future implications for the class. An example of an un insightful model response is "Line number: 25, Segment: "Five feet wide equals 40 square what?"; Reason: The teacher is emphasizing the importance of units in measuring area and prompting the student to include the unit "feet" in their answer." This response is not insightful, because the interpretation is obvious from just this line of the transcript.

- Not insightful  
 Somewhat insightful  
 Insightful

(Optional) Comments: eg. why is this insightful or not insightful?

Note: The "Continue" button will be disabled until you've indicated your ratings on the 3 dimensions.

CONTINUE

Figure 38: Human interface for evaluating the highlights (good examples) and missed opportunities (bad examples) on MQI observation items generated by the model.

## Evaluating Model Suggestions for Eliciting Student Mathematical Reasoning

### Instructions

**Setup:** You will be given a snippet of a classroom transcript and feedback from the model. The model feedback contains 2 suggestions for the teacher on how the teacher could elicit more student mathematical reasoning in the classroom. We define student mathematical reasoning broadly as students asking questions, engaging in mathematical discourse with their teacher or peers, and providing explanations such as justifying their answers.

**Task:** You will be asked to evaluate the quality of the model suggestions along 4 dimensions described below. Some of these dimensions include evaluating whether the suggestions draw on events that actually take place in the transcript or whether the suggestions are useful for the teacher.

Current progress: 0 % completed, 0 / 20

### Transcript #2776, Suggestion #1 on eliciting student mathematical reasoning

9. teacher: So you can do that. Three divided by 6 is going to be again, 0.5. So again we know  $3/6$  equals a half and it's equal to 25 or 0.5. So put it on your table. Put it on your table. Now do you see another one that we can take the numerator and divide it by the denominator and again come out with 0.5 – who can give me another one? Student N?

10. student: Four divided by 8.

11. teacher: So I look at my four as the numerator, I look at my 8 and I'm coming right over here. What am I gonna put in there? 0.5. So I'm putting here's my 4 and then I move my finger over to the 8 and again  $4/8$  is equal to a half so we'll put 0.5 here too. Do I see a pattern going on here?

12. teacher: So 0.5 would be like a half wouldn't it? Do you see another numerator divided by a denominator and you might get a 0.5 again. Who sees another one? Who sees another one? Student M which one?

13. student: Five divided by 10.

14. teacher: Here's my five on here, here's my 10, so I'm gonna come right down and what am I gonna put in there? Five divided by 10 is 0.5. Now if you do that on your calculator once again, if I took five, here's my divided by sign – divided by 10 equals 0.5 again. Any more that we might see? An equivalent fraction that's equal to an equivalent decimal? Student E?

15. student: Six divided by 12.

16. teacher: So I'll come right down. What am I gonna put there? 0.5. Do we know that all these fractions are equal to – that are a half are equal to 0.5? Yes, absolutely. So let's do a little tricky one. I'm gonna point to – and I'll look on the overhead. If I do this, if I take my numerator 1 and I divide it by 3, what is the decimal going to be? 1 divided by 3. Let me do it on my calculator too. If I take a 1 divided by 3 equals – what's it equal, class?

17. teacher: But just for our use we will put 1 divided by 3 and we will put 0.333. That stands for  $1/3$ . That stands for  $1/3$ . So do you all see that on your calculators?

18. teacher: So we could almost say that 0.333 would be equal to what is this, the tens place, the hundreds place and the thousandths place? So we could say that's  $333/1000$  which would look like this over 1000. So that would be  $1/3$ . So these are equivalents, these fractions that we're doing that we see are equivalent to the decimals. They are worth the same. So what you're going to do right now, and we're not going to do some certain ones here. We're not gonna do number 7, we're not going to do 9 and we're not gonna do 11 today. Then we're going to see certain patterns. So let's take a look. Let's start doing some of the and remember let's see what you can come up with. So if we did 1 the numerator divided by 1 the denominator, what's our number?

19. student: One.

20. teacher: So you put the 1 in the first box and then 2 divided by 1 is what?

21. student: Two.

22. teacher: You're putting them in the box, each of the squares where they belong. Do you see what I'm doing everybody? Are you all doing that on your paper? Then we see 3 divided by 1 is what?

### Model prompt

Provide **suggestions** for the teacher on how the teacher could elicit more student mathematical reasoning in the classroom. We define student mathematical reasoning broadly as students asking questions, engaging in mathematical discourse with their teacher or peers, and providing explanations such as justifying their answers.

### Model suggestion #1

Line number: 7, Segment: "So I'm gonna look at my calculator and I'm gonna press 3 then the divided by sign, 6 equals – what's that gonna equal?", Suggestion: Instead of simply asking for the answer, ask students to explain the steps they took to arrive at the answer and to justify their reasoning.

### Evaluate the model's suggestion

#### Rate how **relevant** of the model's suggestion is to eliciting student mathematical reasoning

**Definition of relevance:** The model response is connected to eliciting more student mathematical reasoning in the classroom.

**Examples:** An example of a *relevant* model response is: "I would ask the students to explain their reasoning for their answer." An example of an *irrelevant* model response is "The teacher should control the class better"; this suggestion is connected to classroom management, not eliciting student mathematical reasoning.

- Not relevant
- Somewhat relevant
- Relevant

(Optional) Comments: eg. why is this relevant or not relevant?

#### Rate how **faithful** of an interpretation the model suggestion is

**Definition of faithfulness:** The model response has the right interpretation of the events that occur in the classroom transcript. We evaluate along this dimension because the model sometimes can hallucinate or misinterpret the events in the transcript when providing suggestions.

**Examples:** An example of a *faithful* model response is "Line number: 22, Segment: "Eight what?". Suggestion: The teacher is trying to ask for the units of the answer, but they could fully formulate their question as "What are the units of the answer?". An example of an *unfaithful* model response is "Line number: 3, Segment: "[inaudible]". Suggestion: The student is rushing through their homework" even though there is no evidence of rushing in the transcript.

- Not faithful
- Somewhat faithful
- Faithful

(Optional) Comments: eg. why is this faithful or not faithful?

#### Rate how **actionable** the model suggestion is

**Definition of actionability:** The model suggestion is actionable if it is a focused suggestion that the teacher can easily translate into practice to improve their teaching and encourage student mathematical reasoning.

**Examples:** An example of an *actionable* model response is "Line number: 22, Segment: "Eight what?". Suggestion: The teacher is trying to ask for the units of the answer, but they should fully formulate their questions, such as "What are the units of the answer?". An example of an *unactionable* model response is "Line number: 22, Segment: "Eight what?". Suggestion: The teacher should ask more open-ended questions".

- Not actionable
- Somewhat actionable
- Actionable

(Optional) Comments: eg. why is this actionable or not actionable?

#### Rate how **redundant** the model suggestion is

**Definition of redundancy:** The model suggestion is redundant if it is a suggestion that the teacher is already doing in the transcript.

**Examples:** An example of a *redundant* model response is "Line number: 22, Segment: "Eight what?". Suggestion: The teacher should ask more questions". An example of a *non-redundant* model response is "Line number: 22, Segment: "Eight what?". Suggestion: The teacher should fully formulate their questions, such as "What are the units of the answer?".

- Redundant
- Somewhat redundant
- Not redundant

(Optional) Comments: eg. why is this redundant or not redundant?

Note: The 'Continue' button will be disabled until you've indicated your ratings on the 4 dimensions.

CONTINUE

Figure 39: Human interface for evaluating the model suggestions.

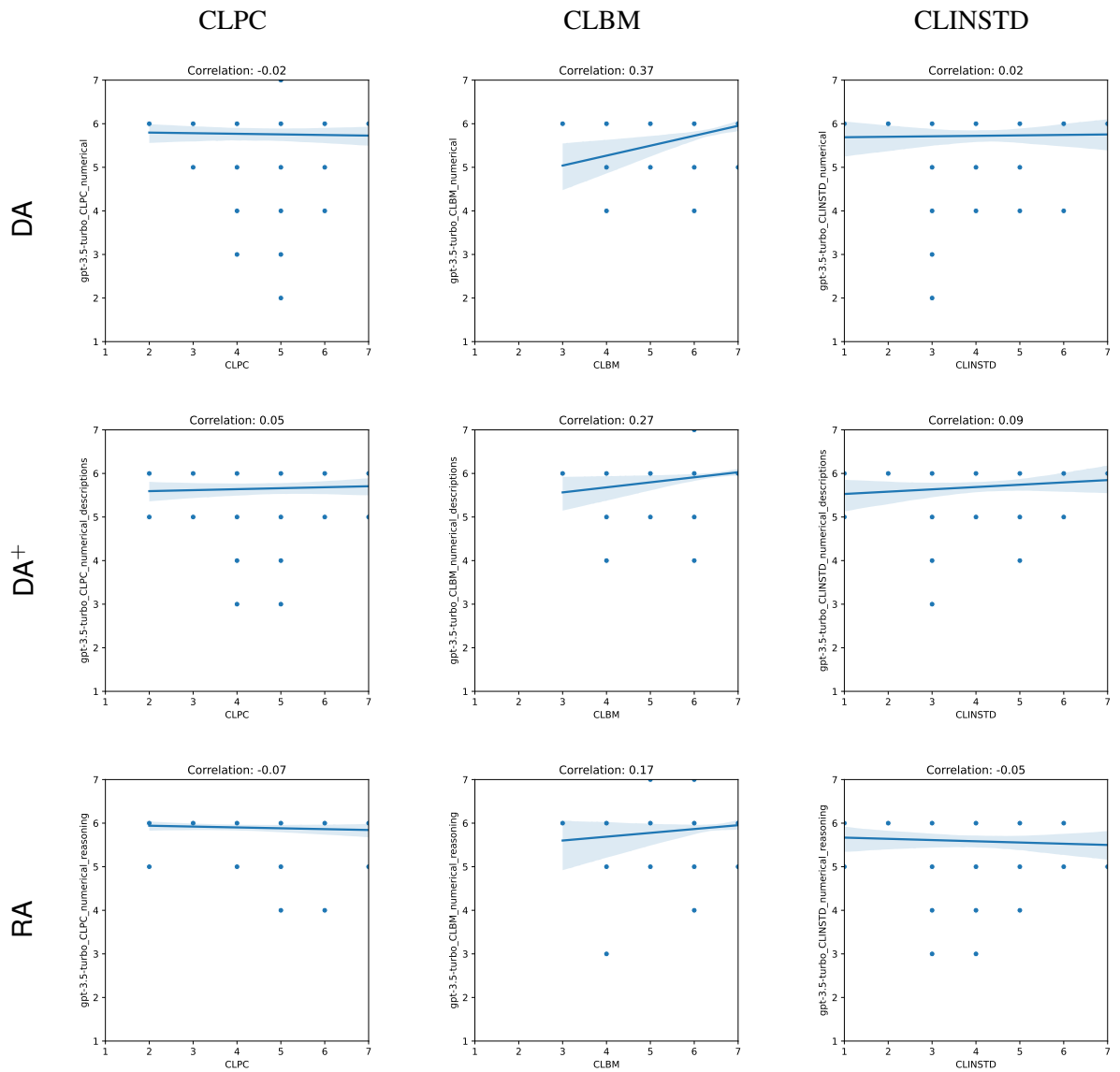


Figure 40: Correlation between CLASS annotations and model predictions.

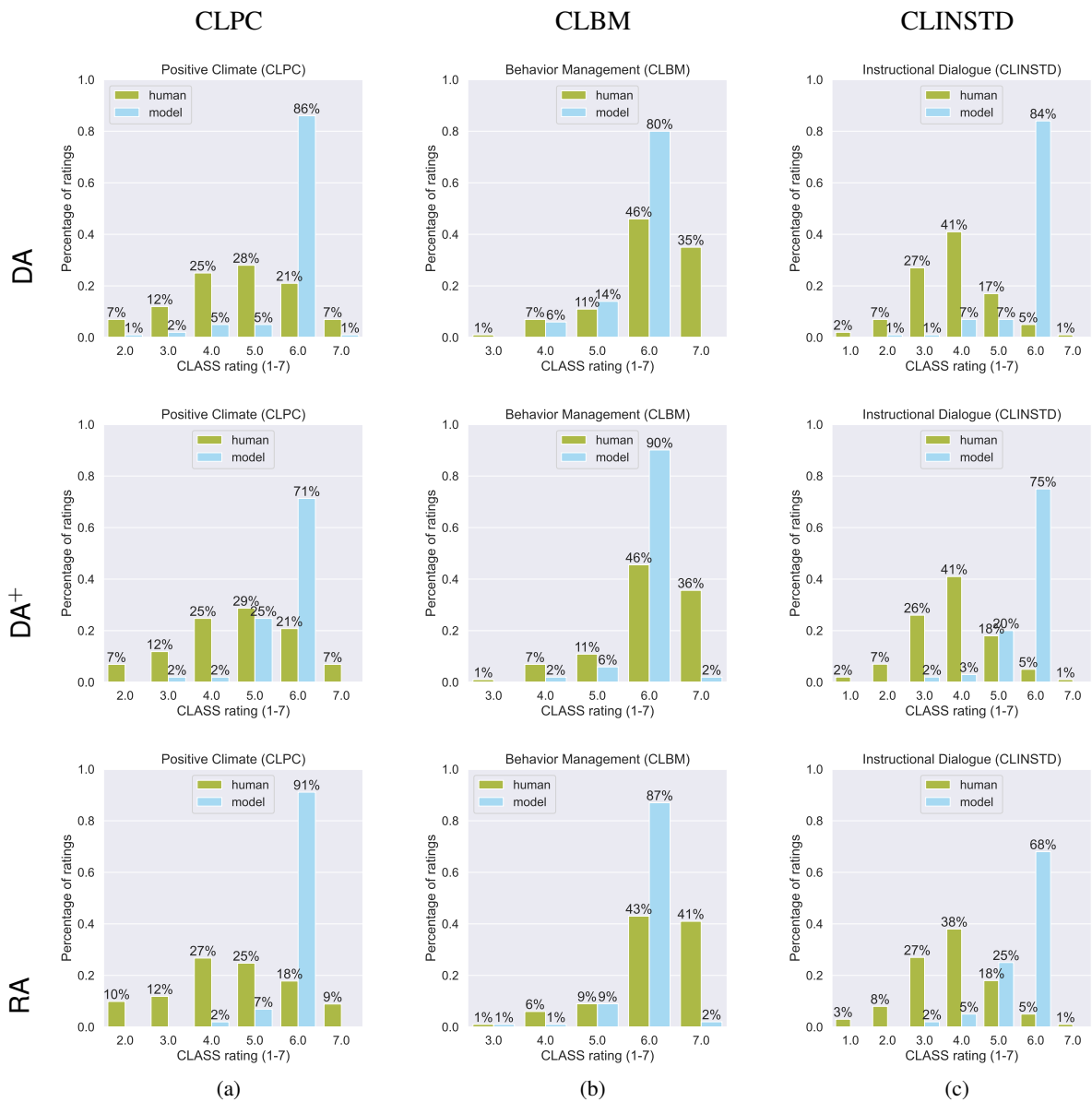


Figure 41: Bar plots comparing CLASS scores from humans vs. ChatGPT model.



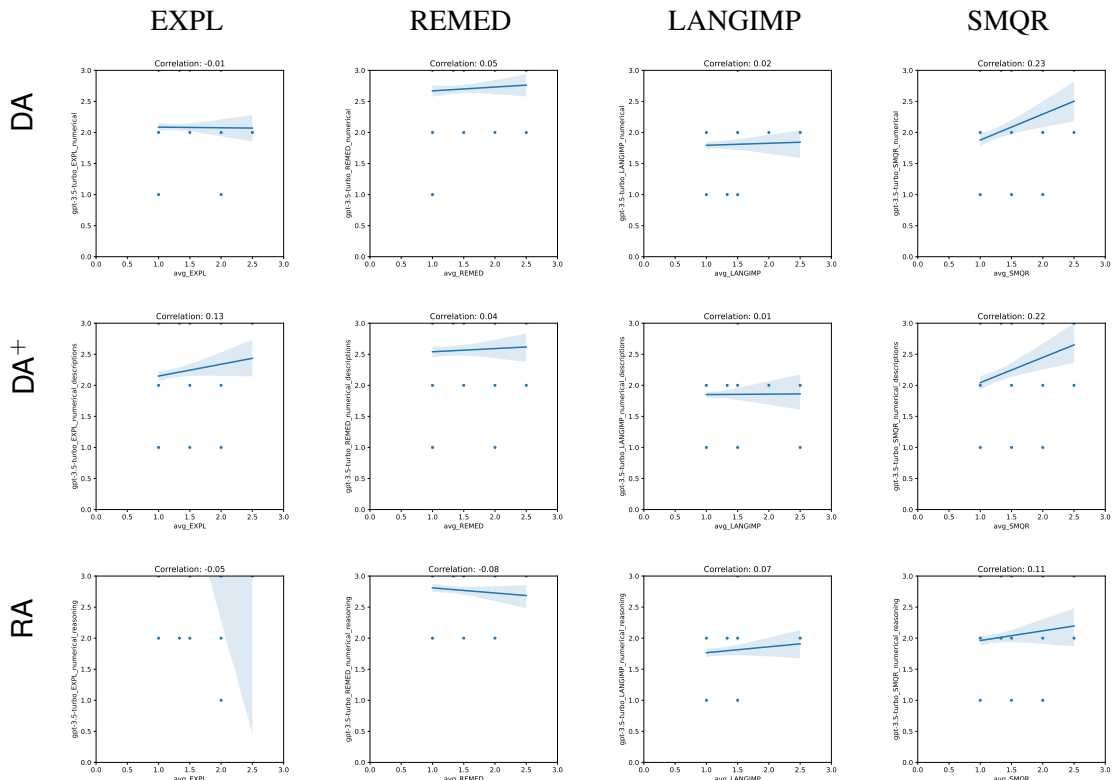


Figure 42: Correlation between MQI annotations and model predictions.

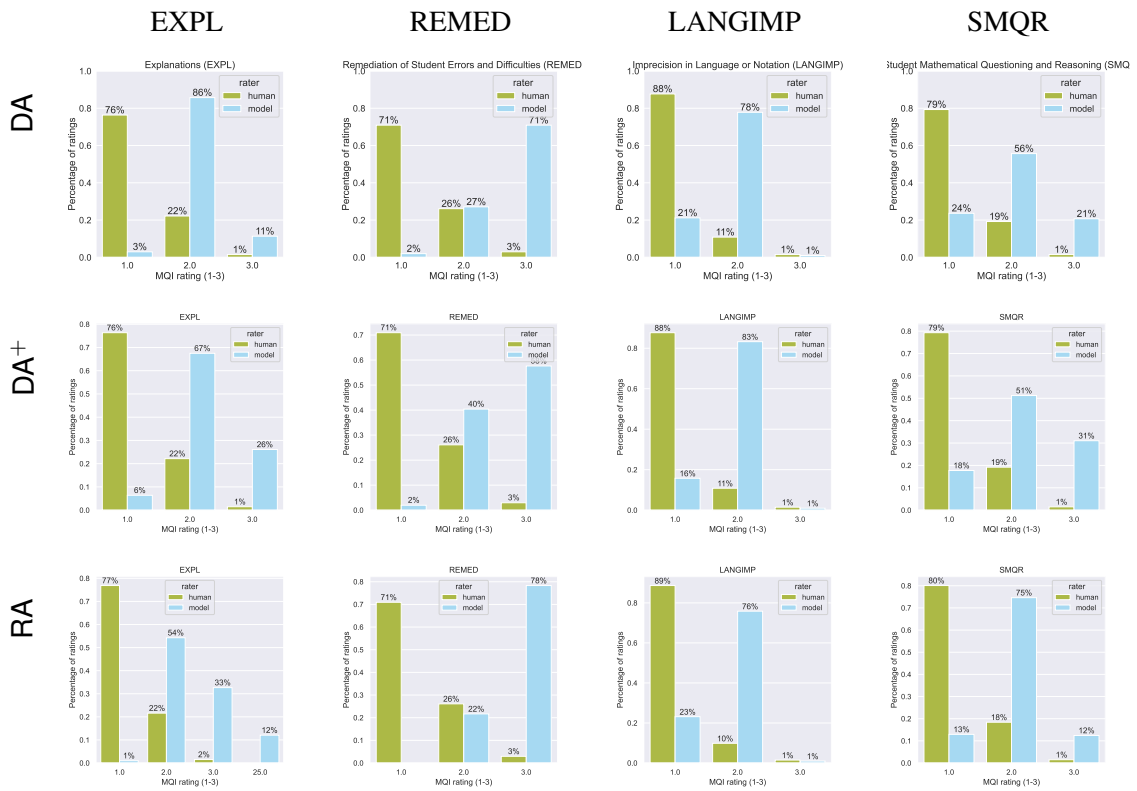


Figure 43: Bar plots comparing MQI scores from humans vs. ChatGPT model.

## Model prompt

Consider the following classroom transcript.

Transcript:

1. teacher: Well, it is division. Take my word for it. I'll write them bigger next time. Raise your hand to tell me, what should I do first? Student H, what are you going to do first?
2. student: What's in the parenthesis.
3. teacher: So you're going to do 30 minus 6 first? And what did you get?
4. student: 23.
5. teacher: Check your subtraction.
6. student: I got 24.
7. teacher: You still got that? What's 10 minus 6?
8. student: 4.
9. teacher: So 30 minus 6 can't be 23. It has to be-
10. multiple students: 24.
11. teacher: Now look. When I saw we do it like this - this is what we did last week. 24 goes right in the middle of those parenthesis. Next I have to bring down what I didn't use. What is 24 divided by 3?
12. student: 8.
13. teacher: Your final answer should have been 8. Does that make sense?
14. multiple students: Yes.
15. teacher: Do you want to do one more example?
16. multiple students: Yes.
17. multiple students: No.
18. teacher: We're going to do one more. I'm going to make it a hard one. You have to follow the steps in the order of operations to get it right. If I do what I think is easiest first, the answer is going to end up different. Here's a hard one. Ready? Copy and complete this one. This has more than three steps. It is 5 times 23 minus 18 plus 7. It looks really hard, but if I follow my steps, it won't be so bad. You have to make sure the parenthesis are in the right spot. For those who are stuck, raise your hand and tell me what should they do first and why. Student M? What are you doing first?
19. student: 5 times-
20. teacher: Is that your order of operations? Is that what you should do first? What's in parenthesis?
21. student: 23 minus 18.
22. teacher: Do 23 minus 18 first. You'll find out. Remember to allow everyone to work. You are quiet. If you're wondering whether you're right or wrong, you'll see in just a minute when we do it together. Where did you write it? Look at the problem that's on the board. It's a different one now. Watch how it looks when you solve it. Student H, here's how it looks when you solve it. The first step you said was to do 23 minus 18, which was what?

The transcript contains many short student responses. Please provide 5 suggestions for the teacher on how the teacher could elicit more student reasoning in the classroom. Student reasoning is counted broadly as students asking questions, engaging in mathematical discourse with their teacher or peers, and providing explanations such as justifying their answers.

Format your answer as:

Advice to the teacher:

1. Line number: <specify line number>, Segment: "<copied from transcript>", Suggestion: <specify advice to the teacher>
2. ...

Advice to the teacher:

## One of the generated model suggestions

Line number: 18, Segment: "What are you doing first?", Suggestion: Encourage the student to explain their thought process and reasoning for choosing a particular step to start with.

Figure 44: Top box: An example prompt for suggestions on eliciting more student mathematical reasoning in the classroom. Bottom box: one of the suggestions generated by the model.

### Evaluation of one of the human annotators

Rating RELEVANCE

Level (low, mid, high): high

Comment: By having the student explain his thought process, the student may have been able to catch his own error rather than having the teacher fix it for him.

Rating FAITHFULNESS

Level (low, mid, high)): high

Comment:

Rating ACTIONABILITY

Level (low, mid, high)): high

Comment:

Rating NOVELTY

Level (low, mid, high)): low

Comment:

Figure 45: One of the human annotator's ratings to the model's response in Figure 44. The human annotators are also shown the transcript the model saw.

# Does BERT Exacerbate Gender or L1 Biases in Automated English Speaking Assessment?

**Alexander Kwako**

University of California, Los Angeles  
akwako@ucla.edu

**Yixin Wan**

University of California, Los Angeles  
elaine1wan@ucla.edu

**Jieyu Zhao**

University of Maryland, College Park  
jieyuz@umd.edu

**Kai-Wei Chang**

University of California, Los Angeles  
kwchang@cs.ucla.edu

**Li Cai**

University of California, Los Angeles  
cai@cresst.org

**Mark Hansen**

University of California, Los Angeles  
markhansen@ucla.edu

## Abstract

In English speaking assessment, pretrained large language models (LLMs) such as BERT can score constructed response items as accurately as human raters. Less research has investigated whether LLMs perpetuate or exacerbate biases, which would pose problems for the fairness and validity of the test. This study examines gender and native language (L1) biases in human and automated scores, using an off-the-shelf (OOS) BERT model. Analyses focus on a specific type of bias known as differential item functioning (DIF), which compares examinees of similar English language proficiency. Results show that there is a moderate amount of DIF, based on examinees' L1 background in grade band 9–12. DIF is higher when scored by an OOS BERT model, indicating that BERT may exacerbate this bias; however, in practical terms, the degree to which BERT exacerbates DIF is very small. Additionally, there is more DIF for longer speaking items and for older examinees, but BERT does not exacerbate these patterns of DIF.

## 1 Introduction

Pretrained large language models (LLMs) present new opportunities for English speaking assessments, yet they are prone to perpetuating and, in some cases, exacerbating social prejudices (Blodgett et al., 2020). In educational assessment, researchers have shown that pretrained LLMs can replicate human scoring, including English speaking assessment, with a high degree of accuracy (Wang et al., 2021). Studies of biases of these automated scoring systems, however, is uncommon (Ormerod, 2022). Considering how widespread

and high stakes English speaking assessments are at both the primary and secondary education levels (Cimpian et al., 2017; Educational Testing Service, 2005), it is imperative that these assessments be fair for all students, regardless of gender or L1 backgrounds. This study addresses the need for deeper analyses of bias in LLM-based automated English speaking assessments.

### 1.1 Bias in English speaking assessment

There are many potential sources of bias in English speaking assessment. We highlight four sources that we believe are most pertinent to the study of gender and L1 biases.

**Human rater bias** Scholarship on implicit bias demonstrates that human judgment is influenced unconsciously by peripheral cues, including speakers' accents (Kang and Yaw, 2021). In the context of English speaking assessment, these biases may lead to unfair scoring without raters even realizing it (Greenwald and Banaji, 1995). Indeed, Winke et al. (2013) reports that human raters are more lenient towards examinees who share the same L1 background. In a summary of research on the biases of raters of L2 English, Lindemann and Subtirelu (2013) reports a strong disconnect between subjective evaluation of speech (e.g. using Likert scales) and more objective measures (e.g. transcription). Although unexplored, implicit bias could also affect examinees based on gender vocal cues.

Research on implicit bias and speech suggests that there may be more bias in the speaking domain, as opposed to other domains, such as writing. By listening to examinees' voices, human raters may be more likely to be influenced by examinees'

accents, triggering implicit bias that affects their judgment during scoring.

**Socio-cultural factors** There are many socio-cultural differences based on gender and L1 that affect English speaking assessment. [Derwing and Munro \(2013\)](#), for instance, discuss how factors like age and conversational opportunities interact with L1 in complex ways. Gender is also a source of variation in L2 English speaking proficiency, although it varies by culture and task ([Denies et al., 2022](#)).

Additionally, cultural differences may interact with item properties. In one highly-publicized study, [Freedle \(2003\)](#) describes how verbal items draw on cultural knowledge that disadvantage minority examinees. It is possible, then, that certain speaking items require an understanding of the context of schooling in the United States, which may be more or less familiar to examinees of different cultural backgrounds, and particularly for those who recently emigrated.

**Curricular differences** [Huang et al. \(2016\)](#) report that curricula vary across countries, and that these differences are a likely source of bias in comparative studies of international assessment. Curricular differences between countries would be particularly salient for examinees who entered into the United States schooling system at a later age.

**Item difficulty** [Dorans and Zeller \(2004\)](#) and [Santelices and Wilson \(2010\)](#) suggest that item difficulty might be related to guessing behavior, which in turn produces bias related to examinees' overall proficiency. Given that speaking is a difficult aspect of L2 language acquisition ([Brown et al., 2000](#)), it is possible that examinees who are less fluent are able to guess their way through non-speaking items, yet struggle with speaking items.

## 1.2 LLMs may exacerbate social biases

Studies have revealed that pretrained LLMs can propagate and, in some cases, amplify negative stereotypes of marginalized groups ([Blodgett et al., 2020](#)). Because LLMs are pretrained on large corpora of text largely scraped from the web, societal biases in these texts become embedded in the LLMs. These biases may surface in downstream applications, such as machine translation ([Stanovsky et al., 2019](#)) and sentiment analysis ([Kiritchenko and Mohammad, 2018](#)).

In English speaking assessment, LLMs are not yet in widespread use. Yet researchers who are

exploring their use typically focus on performance metrics (e.g. accuracy) to the exclusion of biases (e.g. [Wang et al., 2021](#)). Even in the broader field of NLP-based English speaking assessment, analyses of bias are rarely conducted or reported (e.g. [Collier and Huang, 2020](#)). In one rare study, however, [Wang et al. \(2018\)](#) found that their automated scoring system diverged from human raters for several L1 groups.

## 1.3 Differential item functioning

Differential item functioning (DIF) is a specific type of bias commonly examined in educational and psychological assessment ([American Educational Research Association et al., 2014](#)). DIF occurs when “equally able (or proficient) individuals, from different groups, do not have equal probabilities of answering the item correctly” ([Angoff, 1993](#), p. 4).

Although there are many studies of DIF with respect to gender and L1 in large-scale English language assessment, these studies focus on vocabulary, listening, and writing proficiency ([Kunnan, 2017](#)). Very few studies of DIF have been conducted on English speaking proficiency.

## 1.4 Study overview and research questions

This study is designed to analyze gender and L1 biases in L2 English speaking assessment, and to determine if these biases are exacerbated by a pretrained LLM-based automated scoring system. Our data come from a large-scale K-12 English language assessment known as the English Language Proficiency Assessment for the 21st Century (ELPA21; [Huang and Flores, 2018](#)). For our automated scoring model, we use an off-the-shelf pretrained Bidirectional Encoding Representation using Transformers (BERT) model ([Devlin et al., 2018](#)). We focus on BERT because of its seminal status in language modeling, and because it remains a focus of study in English speaking assessment ([Wang et al., 2021](#)). We quantify the amount of bias in human and automated scores by measuring DIF. We first describe specific patterns of DIF in human scores, and then determine whether or not BERT exacerbates DIF.

## 2 Methods

### 2.1 Data

This study draws on data from the English Language Proficiency Assessment for the 21st Century

	Grade Band 2-3			Grade Band 9-12		
	n	%	Avg. Proficiency	n	%	Avg. Proficiency
<b>All</b>	8377	100	0.18 (0.91)	6623	100	0.16 (0.93)
<b>Gender</b>						
Male	4310	51.5	0.13 (0.9)	3648	55.1	0.14 (0.94)
Female	4067	48.5	0.23 (0.92)	2975	44.9	0.2 (0.92)
<b>L1</b>						
Spanish	4205	50.2	0.08 (0.85)	3481	52.6	0.23 (0.92)
Marshallese	692	8.3	-0.0 (0.86)	891	13.5	-0.05 (0.75)
Russian	862	10.3	0.28 (0.9)	375	5.7	0.49 (0.86)
Vietnamese	522	6.2	0.41 (0.9)	402	6.1	0.36 (0.93)
Arabic	499	6	0.33 (0.88)	414	6.3	0.06 (0.86)
Mandarin	439	5.2	0.88 (0.89)	203	3.1	0.44 (1.02)
Hindi	416	5	0.75 (0.82)	185	2.8	0.67 (0.82)
Mayan	238	2.8	-0.66 (0.88)	258	3.9	-0.84 (0.95)
Persian	295	3.5	-0.05 (1.01)	197	3	-0.07 (0.94)
Swahili	209	2.5	0.22 (0.87)	217	3.3	0.04 (0.93)

Table 1: Sample descriptive statistics in aggregate ("All") and disaggregated by gender and L1.

(ELPA21), a consortium involving 7 state education agencies in the U.S. (Huang and Flores, 2018). To maintain confidentiality, certain details regarding test items and examinees are omitted.

Analyses focused on two grand bands (2–3 and 9–12) which corresponded to two tests administered during the 2020–2021 school year. For items in the speaking domain, examinees spoke into a microphone for up to two minutes, after which their responses were sent to human raters who assigned holistic integer scores based on item-specific scoring rubrics. All verbal responses in ELPA21 are currently scored by human raters. Consistent with best practices, raters are trained and monitored over time to ensure consistency (Engelhard, 2002).

## 2.2 Sample design and demographics

The sampling frame included all examinees in grade bands 2–3 or 9–12 who met the following inclusion criteria: answered all three speaking items included in this study; answered at least one item in each of the other three domains; and had gender and L1 demographic information available. To limit the scope of the study, we excluded examinees with disabilities, examinees with non-binary gender, and examinees whose L1 was other than one of the ten L1s analyzed in this study.

From the sampling frame, we sampled 15,000 students.<sup>1</sup> We included all examinees whose L1

<sup>1</sup>The size of our sample was limited, in part, by the cost of

was one of the nine L1 focal groups selected for study (Table 1). The remainder of examinees were randomly sampled from Spanish speakers.

Demographics of grand bands 2–3 and 9–12 are presented in Table 1. Note that there were group differences with respect to overall language proficiency.<sup>2</sup> In both grand bands, male examinees scored slightly lower than female examinees. There was also heterogeneity among L1 groups.

## 2.3 L1 selection

Due to practical limitations, we focused on ten L1 groups. Spanish was the largest L1 group (constituting 82.7% of all examinees in 2020–2021) and, for this reason, served as the reference group. The other nine L1 groups were selected based on the number of examinees available, and with a view to global diversity. See Appendix A for additional details regarding L1 selection and grouping.

## 2.4 Item selection

Speaking items were selected to span a range of response times (i.e., length or quantity of speech). Specifically, for each grand band, we selected one speaking item that was short in duration (i.e., requiring examinees to produce a phrase or simple sentence to answer the prompt), one medium-length item (i.e., requiring 2–3 sentences or a compound

automated transcription.

<sup>2</sup>See Section 2.6 for how language proficiency was computed for examinees.

Item #	Length	Grade Band 2-3			Grade Band 9-12		
		Num. of categories	Avg. seconds	Avg. words	Num. of categories	Avg. seconds	Avg. words
Item 1	short	3	6.4 (4.9)	6.0 (6.5)	4	8.3 (5.0)	11.5 (7.1)
Item 2	medium	5	17.2 (13.3)	25.1 (23.2)	6	14.9 (9.1)	22.8 (16.7)
Item 3	long	6	36.9 (23.1)	51.1 (35.0)	5*	34.7 (18.9)	65.0 (38.4)

Table 2: Item descriptive statistics. Item 3 for grand band 9–12 was re-scaled from a 6-point scale to a 5-point scale. This change was made due to the fact that one group of respondents (Hindi) did not receive any 1s. Combining 1s with 2s helped to improve model convergence.

sentence), and one long item (i.e., requiring 3+ sentences). Table 2 presents the lengths of items 1–3, based on average audio duration (in seconds) and average number of words, for both grand bands. To increase comparability between grand bands, our selection of items also took into consideration item type and item information.

## 2.5 Automated Transcription

Automated transcripts were generated using Amazon Web Services, during October 7–12 and November 14–16, 2022. Default transcription settings were used, with output language set to “en-US.” Amazon provides multiple transcripts by default; the most probable transcripts were selected for analyses.

We conducted an analysis of transcription accuracy and bias of Amazon’s automated transcription service, reported in detail in Kwako (2023). Findings pertinent to the present study are reproduced in Appendix B

## 2.6 Differential item functioning

As discussed in Section 1.3, DIF occurs when there are group differences, conditional on unbiased proficiency estimates. The unbiased proficiency estimate,  $\theta$ , is referred to as the *matching criterion*. In this study, the matching criterion is examinees’ non-speaking English language proficiency (see Section 2.9 for how non-speaking English proficiency was computed). By excluding speaking items, we ensured that estimates of  $\theta$  were not contaminated by the same type(s) of bias under examination. To compare examinees’ of similar  $\theta$ , the sample was divided into ten strata based on which quantile of the standard normal distribution their non-speaking English proficiency resided.

The majority group is referred to as the *reference group*; and the minority group is referred to as the *focal group*. For gender, the reference group was

male (and the focal group was female); for L1, the reference group was Spanish (and the nine focal groups are listed in Table 1).

## 2.7 DIF effect sizes

As summarized by Michaelides (2008), a common method to evaluate DIF for ordinal items is based on the standardized mean difference (SMD) between reference and focal groups (Dorans and Kulick, 1986).<sup>3</sup> SMD is calculated as follows:

$$\sum_j \frac{N_{F,j}}{N_{F..}} \frac{\sum_u N_{Fuj} u}{N_{F,j}} - \sum_j \frac{N_{R,j}}{N_{R..}} \frac{\sum_u N_{Ruj} u}{N_{R,j}}$$

where  $N_{Fuj}$  is the number of examinees in the focal group  $F$  whose  $\theta$  puts them in stratum  $j$ , and who received score  $u$  on the item in question. Multiplying this quantity by  $u$ , and dividing by the number of examinees in the focal group in stratum  $j$ , yields the expected score for the focal group. A similar procedure is followed for the reference group. Before taking the difference, the expected scores are weighted by the proportion of examinees in the focal group in stratum  $j$ .

The effect size,  $z$ , is the ratio of SMD to the standard deviation (pooled between the two groups).<sup>4</sup> Intuitively,  $z$  represents how much the focal group outperforms the reference group, among examinees of similar proficiency, in units of standard deviation.

What counts as a large or small effect size is based on a system originally proposed by Zwick et al. (1993) and is used by the Educational Testing

<sup>3</sup>Instead of using the Mantel test (Mantel, 1963), our significance tests were based on bootstrap sampling distributions and B-H adjusted  $p$ -values, described in Sections 2.10 and 2.11, respectively.

<sup>4</sup>Ormerod et al. (2022) refer to this effect size as  $z$ , a convention we follow.

Service and other educational assessment organizations. Generalizing the system to ordinal items, Allen et al. (2001) designate items as having strong DIF if  $z$  is greater than or equal to 0.25. Items have weak DIF if  $z$  is less than 0.17. And items have moderate DIF if  $z$  is between 0.17 and 0.25.

**Absolute effect size** For certain research questions, the primary interest was not in determining the *direction* of DIF (i.e., which groups are advantaged or disadvantaged), but only in quantifying the *magnitude* of DIF. To address these questions, we based our analyses on the absolute value of  $z$ ,  $z_{abs} = |z|$ . We also refer to this metric as the absolute effect size or absolute DIF.

**Differences between effect sizes** We also computed differences in effect sizes (i.e. between human and automated scores, between items, and between grade bands). In each of these comparisons, we were interested not in  $z$  or  $z_{abs}$ , but in first-order differences. We refer to these quantities as  $\Delta z = z_i - z_j$ , and  $\Delta z_{abs} = |z_{abs,i} - z_{abs,j}|$ , where  $i$  and  $j$  represent two different effect sizes. In research questions 2 and 3, we also examined second order differences,  $\Delta\Delta z_{abs} = |\Delta z_{abs,i} - \Delta z_{abs,j}|$ .

## 2.8 Aggregate DIF metrics

Aggregating DIF effect sizes allowed us to make more general claims about DIF. Analysis of DIF typically revolves around pairwise comparisons at the item level. This fine-grained level of analysis, however, is not suited for making general claims about DIF. To make more general claims (e.g., across multiple items or focal groups) we report *overall* DIF and *factor* DIF.

**Overall DIF** To evaluate DIF across items, we computed  $z$  based on examinees' summed score (i.e. summed across all items of interest). That is, for grand bands 2–3 and 9–12, we added examinees' responses to items 1–3, and computed  $z$  according to the procedure outlined in Section 2.7. Since  $z$  is in units of standard deviation, it is unaffected by differences in items' scales, and thus generalizes well to summed score.

**Factor DIF** Analyses of DIF are usually localized to pairwise comparisons involving one focal group and the reference group. For factors containing more than one focal group, however, we were interested in evaluating DIF for the factor as a whole. To evaluate DIF for the entire factor, we took an unweighted stratified mean of all pairwise comparisons,  $\bar{z}_{abs} = \frac{1}{p} \sum z_{abs,i}$ , where  $p$  is the number of

focal groups. Note that in the case where there is 1 focal group,  $\bar{z}_{abs}$  reduces to  $z_{abs}$ .

## 2.9 Non-speaking English proficiency

Examinees' non-speaking English proficiency was used as the matching criterion in DIF analyses. Non-speaking proficiency was inferred from examinees' responses to test items in non-speaking domains (i.e. listening, reading, and writing). Items were modeled using an Item Response Theory (IRT) framework (Cai et al., 2016), consistent with modeling choices used in production. One difference, however, was that we modeled non-speaking items as a unidimensional construct because (1) it simplified interpretation of the matching criterion, since we were interested in non-speaking proficiency as a whole rather than individual domains, (2) it yielded smaller margins of error, and (3) model fit was in an acceptable range for both grade bands, based on limited-information fit statistics and Tucker-Lewis (non-normed) fit indices (M2 RMSEA  $\leq$  .03 and M2 TLI  $\geq$  .96).

## 2.10 Statistical Estimation

To compute confidence intervals and  $p$ -values, we used a simple bootstrap procedure (Efron and Tibshirani, 1994). Examinees were resampled within grand band, gender, and L1 groups, as these characteristics were central to our study design. Statistics were calculated from 1,000 bootstrapped samples. Confidence intervals were determined from .025 and .975 quantiles for each estimate.  $p$ -values were determined by assuming a normal distribution and taking the minimum of a two-sided quantile of the CDF evaluated at 0.

## 2.11 p-value adjustments

We controlled false discovery rate at the nominal level of .05 using the Benjamini-Hochberg (B-H) technique (Benjamini and Hochberg, 1995). We use the term “statistically significant” (or simply “significant”) when an estimated  $p$ -value is below the B-H adjusted  $p$ -value. In practical terms, statistical significant means that we place an upper bound of .025 on “the probability of being erroneously confident about the direction of the population comparison” (Williams et al., 1999, p. 43).

## 2.12 BERT modeling

Six separate classification models were trained for each of the items analyzed in this study. Cross-entropy served as the loss function. The maxi-



imum number of input tokens depended on the item length: We set the cutoff at two standard deviations above the mean number of tokens for each item. We used the pre-trained uncased BERT base model provided by Huggingface (Wolf et al., 2020). Modeling and training were scripted using Pytorch (Paszke et al., 2019) in Python 9.3.12 (Python Software Foundation, 2022). We explored several possible models with differing hyperparameters as a part of a previous pilot study (Kwako et al., 2022).

### 2.13 BERT training

Data were split 1:1 into testing and training sets.<sup>5</sup> Testing and training sets were split so as to maintain equal proportions of examinees by gender and L1.

Based on a smaller-scale study, we selected learning rates of 1e-6 for BERT layers and 2e-6 for classification heads (Kwako et al., 2022). To slow down overfitting, all but the last attention layer and classification head were frozen during training. Models were trained for 10 epochs, and the epoch with the lowest test loss was selected as the final scoring model for each item.

BERT models nearly achieved parity with human raters for items 1 and 2, and outperformed human raters for item 3. See Appendix C for details regarding the performance of each of the six BERT models in terms of accuracy, correlation, and quadratic weighted kappa (QWK).

## 3 Results

### 3.1 BERT increases DIF for L1

Overall, BERT-based automated scores increased DIF (to a very small degree) with respect to L1 in grade band 9–12. Although this difference was visible across all items in grade band 9–12, item 3 had the largest difference between human and automated scores.

**Overall DIF of human scores** Results revealed a moderate amount of DIF in human ratings based on examinees’ L1 in grade band 9–12. This result is visualized in Figure 1, which shows a gray bar (representing human scores) extending into the yellow (“moderate” DIF) region of the chart ( $z_{abs} = .196, CI_{95\%} = [.170, .222], p = 5.4 \cdot 10^{-48}$ ). Additionally, there was non-zero DIF based on L1 in grade band 2–3, and non-zero DIF based on gender

in grade band 9–12; however, the effect sizes of these quantities were weak.

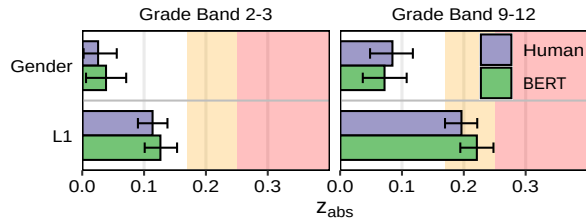


Figure 1: Estimates of overall DIF. Error bars indicate 95% confidence intervals. Yellow shaded regions correspond to moderate DIF, and red shaded regions correspond to strong DIF.

**Human vs. BERT overall DIF** Overall DIF of automated scores was highly similar to human scores. As seen in Figure 1, green bars (representing BERT scores) are nearly commensurate with gray bars (representing human scores), with mostly overlapping 95% confidence intervals. Yet, there was significantly more DIF in BERT scores compared to human scores with respect to L1 in grade band 9–12 ( $\Delta z_{abs} = .025, CI_{95\%} = [.011, .039], p = 3.3 \cdot 10^{-4}$ ). In practical terms, however, an effect size of 0.025 standard deviations is very small.

**Human vs. BERT individual item DIF** In addition to overall DIF, we examined DIF of each individual item. Figure 2 presents DIF of human and automated scores, for gender and L1, across items 1–3, for each grade band. Human and automated scores are again quite consistent. For grade band 9–12, L1 DIF tended to be higher across all items; however, only item 3 reached statistical significance ( $\Delta z_{abs} = .032, CI_{95\%} = [.010, .055], p = 3.3 \cdot 10^{-3}$ ). Again, an effect size of .032 standard deviations is very small.

### 3.2 DIF increases with item length

Based on human rater scores, longer speaking items tended to exhibit more DIF than shorter speaking items. Automated scores did not exacerbate this trend.

By design, item 3 was longer than item 2, which in turn was longer than item 1. Figure 2 shows that, in general, item 3 had more DIF than item 2, which in turn had more DIF than item 1. Table 3 presents the specific values of  $\Delta z_{abs,ij}$ , based on human rater scores, for all three item comparisons. For example, in grade band 9-12, the difference in DIF between items 1 and 2, based on human rater

<sup>5</sup>We set aside a larger percentage of data for testing (50% as opposed to the conventional 20%) because we required a more robust calculation of DIF in the testing set for a related study on debiasing (Kwako, 2023).

Factor	Grade Band 2-3			Grade Band 9-12		
	2 - 1	3 - 1	3 - 2	2 - 1	3 - 1	3 - 2
Gender	.012 [-.030, .051]	.010 [-.029, .049]	-.002 [-.042, .039]	.065 * [.021, .110]	.078 * [.031, .116]	.013 [-.032, .055]
L1	.046 * [.009, .085]	.053 * [.010, .093]	.006 [-.035, .046]	.087 * [.043, .130]	.184 * [.139, .226]	.097 * [.056, .138]

Table 3: Differences in DIF between longer and shorter items, within each grade band, based on human ratings. "\*" indicates that an estimate is statistically significant using B-H adjusted  $p$ -values. 95% confidence intervals are presented in square brackets.

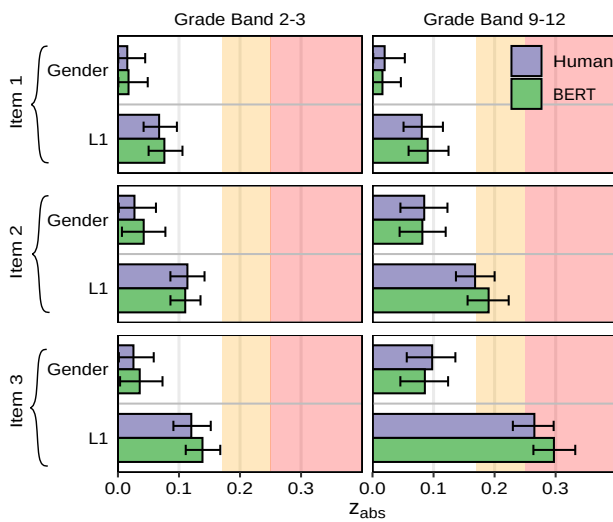


Figure 2: Estimates of DIF for each of the 3 speaking items. Error bars indicate 95% confidence intervals. Yellow shaded regions correspond to moderate DIF, and red shaded regions correspond to strong DIF.

scores (i.e., the gray bars in Figure 2), with respect to L1, was  $\Delta z_{abs,21} = .087$ . That is, item 2 had .087 more standard deviations of DIF compared to item 1. Using B-H adjusted  $p$ -values, this is a statistically significant difference. As indicated by asterisks in Table 3, many (but not all) between-item  $\Delta z_{abs,ij}$  were statistically significant.

Although longer items tend to have more DIF, this general trend was not uniformly consistent across factors and grand bands. Specifically, the trend was less consistent for gender: There were no statistically significant differences in grade band 2–3; and in grade band 9–12, item 3 did not have more DIF than item 2 at a statistically significant level. Additionally, for grade band 2–3, item 3 did not have significantly more DIF than item 2.

In order to determine if item-item differences were exacerbated by automated scoring, we computed second-order differences,  $\Delta\Delta z_{abs}$ . None of these values, however, were statistically significant. We conclude that the pattern of longer items producing more DIF is consistent for both human and automated raters.

### 3.3 DIF is higher for older examinees

In general, there was more DIF for older examinees (in grade band 9–12) compared to younger examinees (in grade band 2–3). Automated scores, however, did not exacerbate this trend.

There was significantly more DIF in grade band 9–12, compared to grade band 2–3, in terms of both gender and L1. This trend can be seen clearly in Figure 1. Based on bootstrapped estimates for gender,  $\Delta z_{abs} = .059$  ( $CI_{95\%} = [.011, .100]$ ,  $p = 4.9 \cdot 10^{-3}$ ); and for L1,  $\Delta z_{abs} = 0.082$  ( $CI_{95\%} = [0.047, 0.120]$ ,  $p = 3.8 \cdot 10^{-6}$ ).

When we examine individual items, this trend is present for items that are medium-length or longer (items 2 and 3) but not for short items (item 1). Visually, this can be seen in Figure 2. The  $\Delta z_{abs}$ , based on human ratings, are presented in Table 4. For example, in item 1, the difference between DIF observed in grade band 2-3 versus grade band 9-12 is  $\Delta z_{abs} = .013$ , with respect to L1, which is not a statistically significant difference. In items 2 and 3, however, the differences between grade band 2-3 and 9-12 are much larger ( $\Delta z_{abs} = .054$  and  $\Delta z_{abs} = .145$ , respectively).

In order to determine if differences between grand bands were exacerbated by automated scoring, we computed second-order differences,  $\Delta\Delta z_{abs}$ . None of these values, however, were statistically significant. We conclude that the trend of greater DIF in older examinees was consistent for both human and automated raters.

Factor	Item 1	Item 2	Item 3
Gender	.005 [-.033, .042]	.058 * [.011, .105]	.072 * [.019, .118]
L1	.013 [-.029, .057]	.054 * [.012, .098]	.145 * [.098, .193]

Table 4: Differences in DIF between grand bands, based on human ratings, for each of the three speaking items. "\*" indicates that an estimate is statistically significant using B-H adjusted p-values. 95% confidence intervals are provided in square brackets.

### 3.4 Severity of DIF depends on L1 and grade band

The direction and magnitude of DIF varied by L1 background, and patterns were generally not consistent across grand bands. Figure 3 depicts the magnitude and direction of DIF for gender and all L1 groups. For grade band 2–3, native speakers of Marshallese and Mayan languages showed evidence of moderate–strong DIF for human and BERT scores. DIF was negative for both L1 groups, indicating that these examinees fared worse on speaking items than their (equally-proficient) Spanish-speaking counterparts.

In grade band 9–12, examinees of nearly all L1 backgrounds fared better than native Spanish speakers. In this case, speaking items tended to disadvantage members of the reference group (i.e. examinees with Spanish L1 backgrounds).

As with preceding analyses, DIF based on BERT scores aligned closely with DIF based on human scores. Although results showed that BERT exacerbated DIF in L1 as a whole (Section 3.1), analyses of individual L1 groups did not reveal any statistically significant differences between human and BERT scores. We also did not find any statistically significant differences between human and BERT scores when examining DIF at the individual item level (Appendix D).

## 4 Discussion

### 4.1 Main findings

Analysis of differential item functioning (DIF) revealed several patterns of biases in L2 English speaking assessment based on human rater scores, some of which biases were exacerbated by BERT-based automated scores. With respect to human scores, we found that there was more DIF for older examinees and for longer items. Based on commonly accepted standards regarding effect size,

there was a moderate amount of overall DIF in grade band 9–12 based on examinees' native language (L1) backgrounds. Automated scores generated by off-the-shelf BERT models closely matched human scores, yet BERT was found to exacerbate overall DIF for grade band 9–12 based on examinees' L1. The degree to which BERT exacerbated this bias, however, was very small.

### 4.2 Causes of DIF

Although our findings do not confirm any causes of DIF, they do allow us to rule out several possibilities.

**Transcription (in)accuracy** Prior research showed that there were discrepancies in transcription accuracy based on speakers' L1 background B. Specifically, automated transcription struggled with speakers of Vietnamese L1 backgrounds in grade band 9–12. Yet given the close correspondence between human and automated scores for all examinees, not just Vietnamese examinees, it appears unlikely that transcription inaccuracies engendered lower or higher scores.

**Implicit bias** Our automated scoring system was based exclusively on transcripts of examinees' speech. No phonic information was used in the automated scoring process. It is notable, then, that there was no mitigation of DIF in automated scores using the text-based BERT model. In other words, removal of acoustic input did not reduce bias. From this, we conclude that examinees with *identical* (transcribed) responses could not have received higher or lower scores, on average, based on gender or L1.

Although text-based automated scores did not mitigate bias, this does not necessarily imply that human raters were unaffected by implicit bias. It is possible, for instance, that examinees with different accents also had different (transcribed) responses, which still affected human raters' judgment.

### 4.3 Limitations

Our analyses were based around one metric of uniform DIF,  $z$ . The benefits of  $z$  are that it is commonly used in practice, it is highly interpretable with well-established effect sizes, and it is easy to aggregate across items and focal groups. One of the drawbacks, however, is that it does not capture non-uniform DIF, and it is not ideal in terms of statistical power (Woods et al., 2013).

Consistent with other analyses of DIF, our study struggles to identify sources of DIF (Zumbo, 2007).

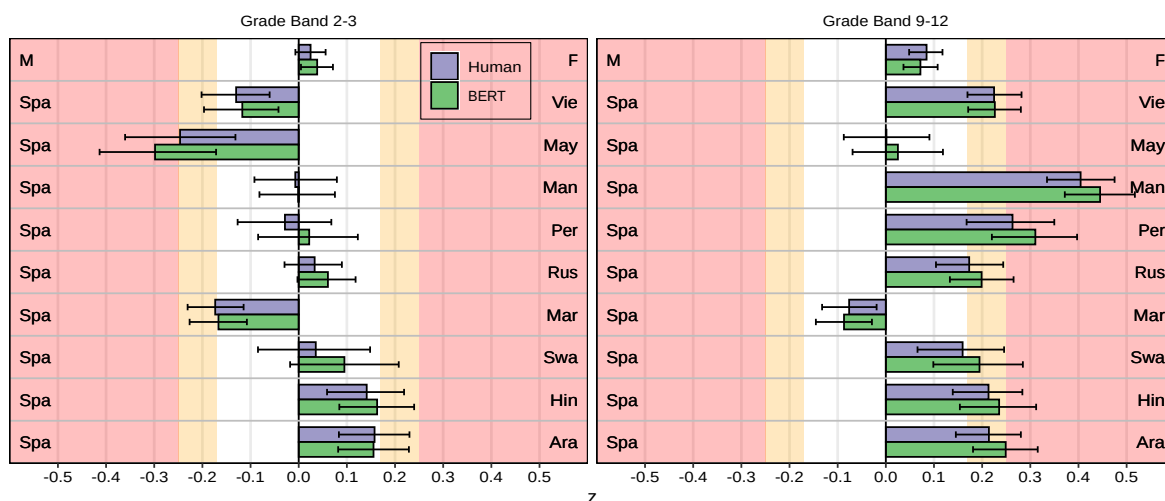


Figure 3: Estimates of direction and magnitude of overall DIF. Error bars indicate 95% confidence intervals. Yellow shaded regions correspond to moderate DIF, and red shaded regions correspond to strong DIF. Reference groups are listed on the left of each chart (M = Male, Spa = Spanish); focal groups are listed on the right (L1 groups are abbreviated by the first three letters). DIF in the positive direction indicates that the focal group is favored.

Although it is outside the scope of this study, a fine-grained analysis of examinees' language, especially based on L1, could provide insight. Additionally, it could be beneficial to explore the possibility of modifying BERT using debiasing techniques (Sun et al., 2019). These techniques could potentially reveal sources of DIF and reduce DIF. Follow-up analyses along these lines of inquiry may be found in Kwako (2023).

## References

Nancy L Allen, John R Donoghue, and Terry L Schoeps. 2001. The naep 1998 technical report. *Education Statistics Quarterly*, 3(4):95–98.

American Educational Research Association, American Psychological Association, and National Council on Measurement in Education. 2014. *Standards for educational and psychological testing*. American Educational Research Association.

William H Angoff. 1993. Perspectives on differential item functioning methodology.

Yoav Benjamini and Yocef Hochberg. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300.

Su Blodgett, Solon Barocas, Hal Daumé, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of "bias" in nlp.

H Douglas Brown et al. 2000. *Principles of language learning and teaching*, volume 4. Longman New York.

Keith Brown. 2005. *Encyclopedia of language and linguistics*, volume 1. Elsevier.

Li Cai, Kilchan Choi, Mark Hansen, and Lauren Harrell. 2016. Item response theory. *Annual Review of Statistics and Its Application*, 3:297–321. Publisher: Annual Reviews.

Joseph R Cimpian, Karen D Thompson, and Martha B Makowski. 2017. Evaluating english learner reclassification policy effects across districts. *American Educational Research Journal*, 54(1\_suppl):255S–278S.

Jo-Kate Collier and Becky Huang. 2020. Test review: Texas english language proficiency assessment system (telpas). *Language Assessment Quarterly*, 17(2):221–230.

Katrijn Denies, Liesbet Heyvaert, Jonas Dockx, and Rianne Janssen. 2022. Mapping and explaining the gender gap in students' second language proficiency across skills, countries and languages. *Learning and Instruction*, 80:101618.

Tracey M Derwing and Murray J Munro. 2013. The development of l2 oral language skills in two l1 groups: A 7-year study. *Language learning*, 63(2):163–185.

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*.

- Neil J Dorans and Edward Kulick. 1986. Demonstrating the utility of the standardization approach to assessing unexpected differential item performance on the scholastic aptitude test. *Journal of educational measurement*, 23(4):355–368.
- Neil J Dorans and Karin Zeller. 2004. Examining freedle’s claims about bias and his proposed solution: Dated data, inappropriate measurement, and incorrect and unfair scoring. *ETS Research Report Series*, 2004(2):1–33.
- Educational Testing Service. 2005. Test and score data summary: 2004-05 test year data test of english as a foreign language.
- Bradley Efron and Robert J Tibshirani. 1994. *An introduction to the bootstrap*. CRC press.
- G Engelhard. 2002. Monitoring raters in performance assessments. *Large-scale assessment programs for all students: Validity, technical adequacy, and implementation*, pages 261–287.
- Roy Freedle. 2003. Correcting the sat’s ethnic and social-class bias: A method for reestimating sat scores. *Harvard Educational Review*, 73(1):1–43.
- Anthony G Greenwald and Mahzarin R Banaji. 1995. Implicit social cognition: attitudes, self-esteem, and stereotypes. *Psychological review*, 102(1):4.
- Becky H Huang and Belinda Bustos Flores. 2018. The english language proficiency assessment for the 21st century (elpa21). *Language Assessment Quarterly*, 15(4):433–442.
- Xiaoting Huang, Mark Wilson, and Lei Wang. 2016. Exploring plausible causes of differential item functioning in the pisa science assessment: language, curriculum or culture. *Educational Psychology*, 36(2):378–390.
- Okim Kang and Katherine Yaw. 2021. Social judgement of l2 accented speech stereotyping and its influential factors. *Journal of Multilingual and Multicultural Development*, pages 1–16.
- Svetlana Kiritchenko and Saif M Mohammad. 2018. Examining gender and race bias in two hundred sentiment analysis systems. *arXiv preprint arXiv:1805.04508*.
- Antony John Kunnan. 2017. *Evaluating language assessments*. Taylor & Francis.
- Alexander Kwako. 2023. *Mitigating Gender and Racial Bias in Automated English Speaking Assessment*. University of California, Los Angeles.
- Alexander Kwako, Yixin Wan, Jieyu Zhao, Kai-Wei Chang, Li Cai, and Mark Hansen. 2022. Using item response theory to measure gender and racial bias of a bert-based automated english speech assessment system. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 1–7.
- Stephanie Lindemann and Nicholas Subtirelu. 2013. Reliably biased: The role of listener expectation in the perception of second language speech. *Language Learning*, 63(3):567–594.
- Nathan Mantel. 1963. Chi-square tests with one degree of freedom; extensions of the mantel-haenszel procedure. *Journal of the American Statistical Association*, 58(303):690–700.
- Michalis P Michaelides. 2008. An illustration of a mantel-haenszel procedure to flag misbehaving common items in test equating. *Practical Assessment, Research, and Evaluation*, 13(1):7.
- Christopher Ormerod. 2022. Short-answer scoring with ensembles of pretrained language models. *arXiv preprint arXiv:2202.11558*.
- Christopher Ormerod, Susan Lottridge, Amy E Harris, Milan Patel, Paul van Wamelen, Balaji Kodeswaran, Sharon Woolf, and Mackenzie Young. 2022. Automated short answer scoring using an ensemble of neural networks and latent semantic analysis classifiers. *International Journal of Artificial Intelligence in Education*, pages 1–30.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, and Luca Antiga. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037.
- Python Software Foundation. 2022. [The python language reference](#).
- Maria Veronica Santelices and Mark Wilson. 2010. Unfair treatment? the case of freedle, the sat, and the standardization approach to differential item functioning. *Harvard Educational Review*, 80(1):106–134.
- Gabriel Stanovsky, Noah A Smith, and Luke Zettlemoyer. 2019. Evaluating gender bias in machine translation. *arXiv preprint arXiv:1906.00591*.
- Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang. 2019. Mitigating gender bias in natural language processing: Literature review. *arXiv preprint arXiv:1906.08976*.
- Xinhao Wang, Keelan Evanini, Yao Qian, and Matthew Mulholland. 2021. Automated scoring of spontaneous speech from young learners of english using transformers. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 705–712. IEEE.
- Zhen Wang, Klaus Zechner, and Yu Sun. 2018. Monitoring the performance of human and automated scores for spoken responses. *Language Testing*, 35(1):101–120.

Valerie SL Williams, Lyle V Jones, and John W Tukey. 1999. Controlling error in multiple comparisons, with examples from state-to-state differences in educational achievement. *Journal of educational and behavioral statistics*, 24(1):42–69.

Paula Winke, Susan Gass, and Carol Myford. 2013. Raters’ L2 background as a potential source of bias in rating oral performance. *Language Testing*, 30(2):231–252.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, and Sam Shleifer. 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.

Carol M Woods, Li Cai, and Mian Wang. 2013. The longer-improved wald test for dif testing with multiple groups: Evaluation and comparison to two-group irt. *Educational and Psychological Measurement*, 73(3):532–547.

Klaus Zechner. 2009. What did they actually say? agreement and disagreement among transcribers of non-native spontaneous speech responses in an english proficiency test. In *International Workshop on Speech and Language Technology in Education*.

Bruno D Zumbo. 2007. Three generations of dif analyses: Considering where it has been, where it is now, and where it is going. *Language assessment quarterly*, 4(2):223–233.

Rebecca Zwick, John R Donoghue, and Angela Grima. 1993. Assessment of differential item functioning for performance tasks. *Journal of Educational Measurement*, 30(3):233–251.

## A L1 Groups

In selecting L1 groups, one of our aims was to represent languages from around the globe. In some cases, this required grouping languages to reach an adequate sample size for statistical analyses. Given the constraints of sample size, we tried to ensure that L1 groups were as geo-historically related to each other as possible (Brown, 2005). The four composite L1 groups in our study were (1) Hindi, (2) Mayan languages, (3) Persian, and (4) Swahili. For simplicity, we refer to composite L1 groups by the predominate language within each group, with the exception of Hindi (in order to remain consistent with a prior study). It would be more accurate, however, to refer to the L1 groups as (1) Indo-Aryan, (2) Indigenous languages of Central and South America, (3) Indo-European languages of the Middle East, and (4) Niger-Congo languages.

The languages within each of the composite L1 groups are presented in Table 5. Note that the names of languages are derived from states’ departments of education, which do not follow the same naming conventions. We made minor changes in compiling the list of languages (e.g. changing “Panjabi” to “Punjabi”).

There is a great deal of heterogeneity within L1 groups, as with gender, and as with all other demographic characteristics. We note that L1 is not synonymous with cultural identity, racial identity, geographic identity, or preferred language. Despite these limitation, in the context of English speaking assessment, we believe L1 is a more relevant construct than, say, conventional racial categories (e.g. White, Asian, Black).

## B BERT Performance Metrics

We conducted an analysis of the accuracy and bias of Amazon’s automated transcription service. The methodology and results of this study are reported in detail in Kwako (2023); however, pertinent aspects of the study are also presented here. Briefly, we evaluated transcription accuracy by computing word error rate (WER), a common metric that represents the number of transcription errors (i.e. insertions, deletions, and substitutions) as a percentage of words in a given utterance. Transcripts generated by Amazon were compared to a set of manually-generated (“ground truth”) transcripts.

Figure 4 presents the WER of automated transcription for grade bands 2-3 and 9-12. Overall, examinees in grand band 2–3 had a higher WER, on average, than examinees in grand band 9–12 (20.5% versus 16.5%, respectively). Note that this level of accuracy is on par with human-human levels of (dis)agreement for L2 English speech, which typically ranges from 15-20% (Zechner, 2009).

There were no statistically significant differences in either grade band with respect to gender. There were also no statistically significant differences in grade band 2-3 with respect to examinees’ L1. Yet in grade band 9-12, examinees’ whose L1 was Arabic had a lower WER (9.1%), on average, compared to other L1 groups. In contrast, examinees whose L1 was Vietnamese had a higher WER (26.3%) than other L1 groups.

As discussed in Section 3.4, there were no statistically significant differences with respect to overall DIF, when comparing human and BERT scores, based on examinees’ L1 groups. Given the close

Language	Grade Band 2-3		Grade Band 9-12	
	n	%	n	%
<b>Hindi</b>				
Punjabi	157	37.7	75	40.5
Hindi	124	29.8	39	21.1
Urdu	65	15.6	35	18.9
Gujarati	46	11.1	30	16.2
Marathi	24	5.8	6	3.2
<b>Mayan languages</b>				
Mayan languages	212	89.1	214	82.9
Q'anjob'al	24	10.1	40	15.5
Quechua	1	0.4	3	1.2
Q'eqchi	1	0.4	1	0.4
<b>Persian</b>				
Persian	209	70.8	97	49.2
Kurdish	76	25.8	87	44.2
Farsi	10	3.4	13	6.6
<b>Swahili</b>				
Swahili	89	42.6	120	55.3
Nuer	37	17.7	28	12.9
Niger-Kordofanian languages	16	7.7	16	7.4
Dinka	19	9.1	11	5.1
Kinyarwanda	7	3.3	19	8.8
Wolof	15	7.2	10	4.6
Fulah	10	4.8	5	2.3
Igbo	7	3.3	5	2.3
Yoruba	3	1.4	1	0.5
Hausa	1	0.5	1	0.5
Akan	2	1	0	0
Shona	2	1	0	0
Chichewa; Chewa; Nyanja	0	0	1	0.5
Kirundi	1	0.5	0	0

Table 5: Languages of composite L1 groups by grand band.

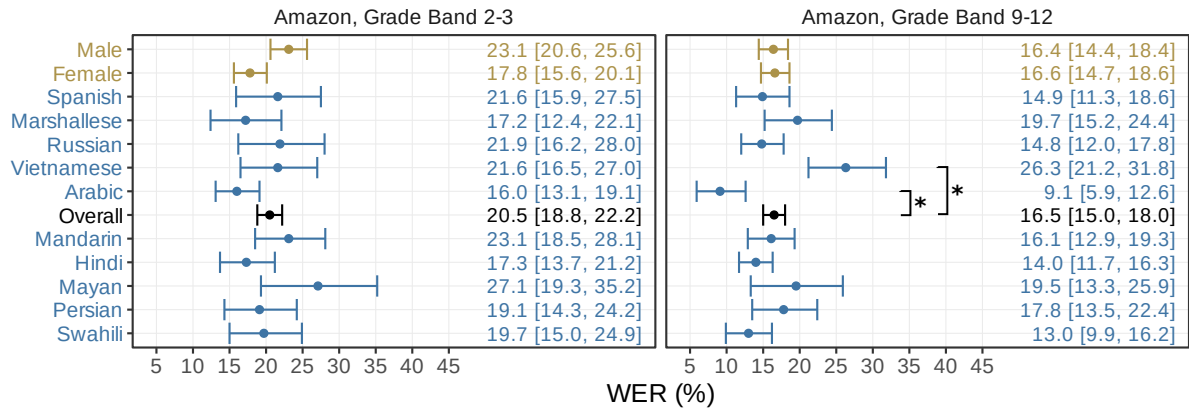


Figure 4: Average word error rate (WER) estimates produced by Amazon’s automated transcription service. Overall WER appear in black, and disaggregated WER appear in gold (gender) and blue (L1); whiskers indicate 95% confidence intervals; brackets with asterisks indicate statistically significant pairwise comparisons.

Item	Grade Band 2-3						Grade Band 9-12					
	Acc.		r		QWK		Acc.		r		QWK	
	H	B	H	B	H	B	H	B	H	B	H	B
1	.911	.896	.793	.713	.792	.713	.929	.904	.920	.895	.920	.895
2	.756	.685	.898	.861	.898	.859	.728	.700	.911	.910	.911	.909
3	.614	.618	.834	.834	.834	.829	.694	.707	.841	.885	.609	.884

Table 6: Performance of off-the-shelf BERT scoring models for items 1–3, compared to human-human agreement, with respect to accuracy, correlation ( $r$ ), and quadratic weighted kappa (QWK). "H" refers to human-human comparisons (i.e. rater 2 compared to rater 1). The number of observations that were scored by two human raters ranged from 1,567–1641 for Grade Band 2–3, and from 1,254–1,293 for Grade Band 9–12. "B" refers to human-BERT comparisons (i.e. BERT compared to rater 1). The number of observations in the testing sets were 4,185 for Grade Band 2–3, and 3,306 for Grade Band 9–12.

correspondence between human and BERT scores, it is unlikely that transcription inaccuracies engendered lower or higher scores.

### C BERT Performance Metrics

Performance metrics of all six BERT models are presented in Table 6. Approximately 10% of all responses were scored by two human raters, independently, which provides the basis for comparisons between human and BERT performance. Off-the-shelf BERT models performed marginally worse for items 1 and 2, but were more consistent than human raters for item 3, across most metrics.

### D Human vs. BERT DIF for each item

Figure 5 presents the magnitude and direction of DIF of items 1-3 for grand bands 2-3 and 9-12, based on gender and all nine L1 focal groups separately.



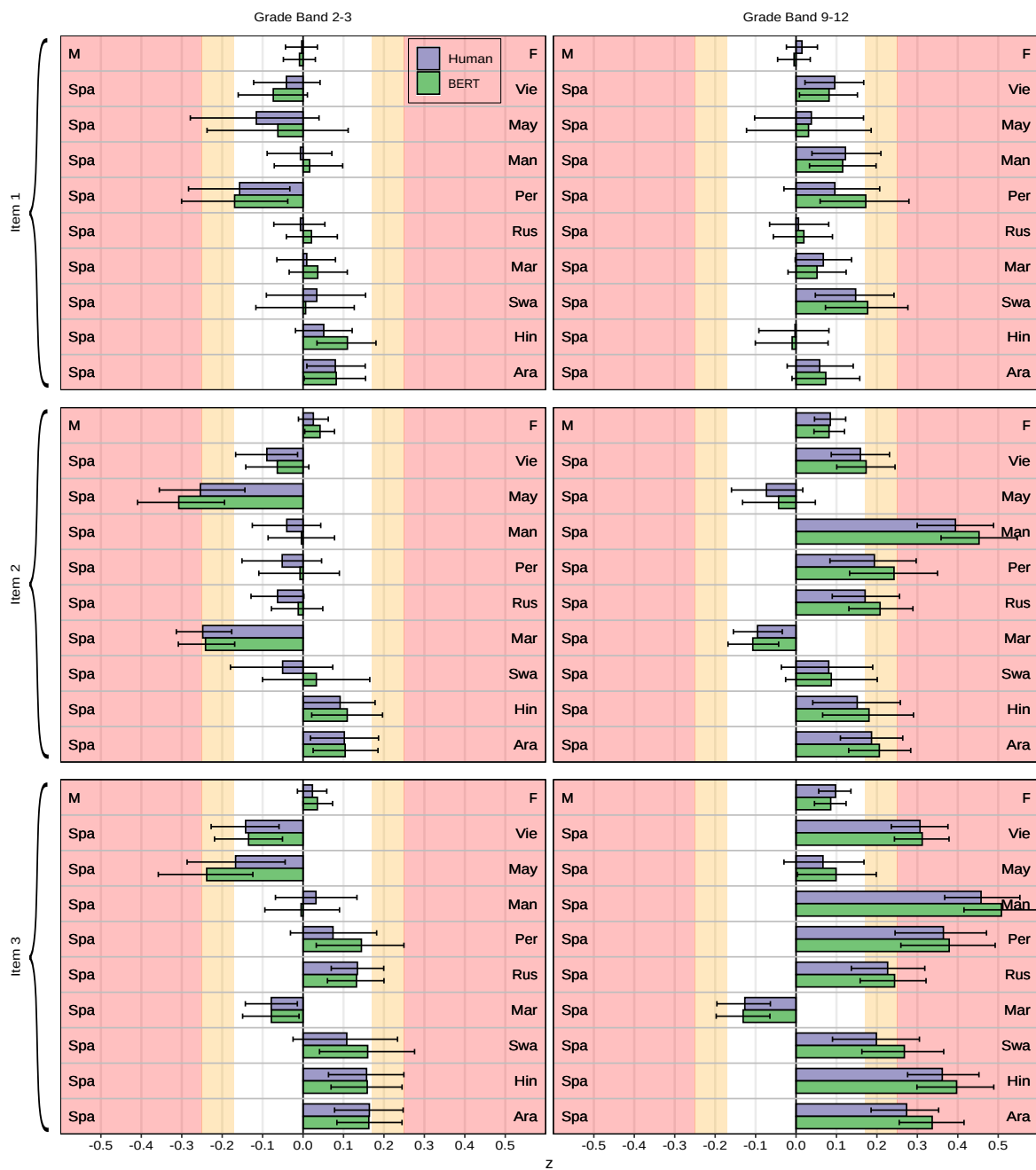


Figure 5: Estimates of direction and magnitude of DIF for each of the three speaking items. Error bars indicate 95% confidence intervals. Yellow shaded regions correspond to moderate DIF, and red shaded regions correspond to strong DIF. Reference groups are listed on the left of each chart (M = Male, Spa = Spanish); focal groups are listed on the right (L1 groups are abbreviated by the first three letters). DIF in the positive direction indicates that the focal group is favored.

# MultiQG-TI: Towards Question Generation from Multi-modal Sources

Zichao Wang\*  
Adobe Research  
jackwa@adobe.com

Richard G. Baraniuk  
Rice University  
richb@rice.edu

## Abstract

We study the new problem of automatic question generation (QG) from multi-modal sources containing images and texts, significantly expanding the scope of most of the existing work that focuses exclusively on QG from only textual sources. We propose a simple solution for our new problem, called MultiQG-TI, which enables a text-only question generator to process visual input in addition to textual input. Specifically, we leverage an image-to-text model and an optical character recognition model to obtain the textual description of the image and extract any texts in the image, respectively, and then feed them together with the input texts to the question generator. We only fine-tune the question generator while keeping the other components fixed. On the challenging ScienceQA dataset, we demonstrate that MultiQG-TI significantly outperforms ChatGPT with few-shot prompting, despite having hundred-times less trainable parameters. Additional analyses empirically confirm the necessity of both visual and textual signals for QG and show the impact of various modeling choices. Code is available at <https://rb.gy/020tw>

## 1 Introduction

Automatic question generation has the potential to enable personalized education experiences for subjects such as reading comprehension at a large scale (Wolfe, 1976; Kokku et al., 2018; Zhang et al., 2022; Kulshreshtha et al., 2022) and improve standardized tests by reducing the costs and the test length (Burstein et al., 2021). Most, if not all, existing question generation (QG) methods operate *only on text*: they take a *textual* paragraph (Wang et al., 2018) or story (Xu et al., 2022) as input and generate a *textual* question. These methods' focus on text-based QG is limiting, because many interesting questions can involve, or be generated from, multiple modalities such as images, diagrams, and tables, in addition to texts (Lu et al., 2022).

\*Work done while at Rice University.

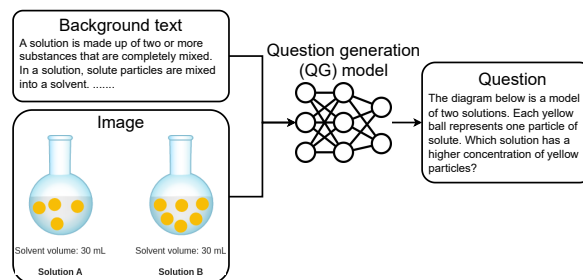


Figure 1: Illustration of our multi-modal question generation (QG) problem. Given a background text and an image, our goal is to develop a model to automatically generate a textual question based on them.

### 1.1 Contributions

In this paper, we conduct, to our knowledge, the *first* investigation into the under-explored problem of *multi-modal* question generation (QG). Specifically, we study the following problem: given multi-modal inputs containing both visual (e.g., an image) and textual (e.g., a textbook paragraph) information, we would like a model to output a textual question based on such multi-modal input. Note that the definition of visual input is very broad, e.g., it can be an image, a diagram, or a table in the image format. Although this multi-modal setting (image and text as input and textual question as output) is only a specific instance of multi-modality (one could consider using audio and video as input to generate questions, or generating questions with images in addition to texts), we argue that our setting is sufficiently broad and educationally meaningful. For example, many science questions ask about scientific phenomena, processes, and relationships commonly described in figures, diagrams, and tables (Talmor et al., 2021; Lu et al., 2022). We believe that our problem setting, illustrated in Figure 1, is an important first step toward more general multi-modal QG.

We propose a novel method, dubbed MultiQG-TI, for generating textual questions from multi-modal inputs of texts and images. The idea is simple: we enable a text-based question genera-

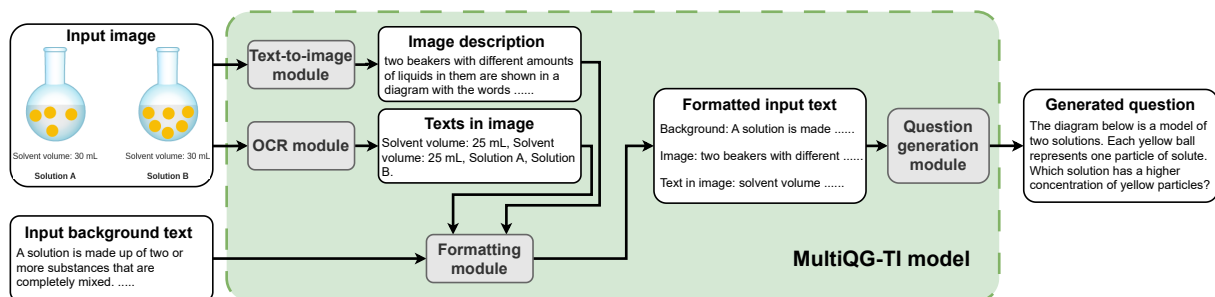


Figure 2: Illustration of the proposed MultiQG-TI methodology.

tor to “see” by feeding it visual information in the form of text. Specifically, we first use an off-the-shelf image-to-text model and an optical character recognition (OCR) model to produce a textual description of the image and extract the texts in the image. We then fine-tune a text-based generative model to generate a question given the input text and the text extracted from the input image. These components are readily available and require no or minimal fine-tuning, making MultiQG-TI easy to use and efficient to train. Figure 2 presents a high-level overview of MultiQG-TI.

We demonstrate MultiQG-TI’s strong performance on the challenging ScienceQA dataset (Lu et al., 2022). For example, MultiQA-TI outperforms models using only texts or only images as input, demonstrating the necessity of including both texts and images as input in QG. MultiQA-TI also significantly outperforms ChatGPT in the few-shot in-context learning setting, demonstrating its competitiveness against much larger models. Finally, we analyze the factors that impact MultiQA-TI’s performance, including the choices of image-to-text models and the sizes of the question generator model. We also provide generation examples to illustrate our method’s strengths and errors.

## 1.2 Related Work

**Question generation (QG) for education.** QG models are often an integral component in personalized learning, intelligent tutoring systems, and assessment platforms to cheaply and scalably generate customized questions for each student (Le et al., 2014; Pan et al., 2019; Srivastava and Goodman, 2021; White et al., 2022). For example, prior research has developed models to generate a variety of questions including those based on fairytales (Xu et al., 2022; Zhao et al., 2022), factual questions (Heilman and Smith, 2010; Wang et al., 2018), and math word problems (Wang et al., 2021; Liu et al., 2021). Despite the rapid progress, most

existing work focuses on *textual-based* QG. The exciting frontier of automatic multi-modal QG remains under-explored.

**Multi-modal processing with text-only models.** Our work is partially motivated by the recent line of work that demonstrate the possibility to use text-only models to perform visual-related tasks by feeding it text descriptors of the visual input. For example, Wang et al. (2022) enable large language models to perform video-related tasks such as event prediction by connecting them with image-to-text models. A few others take a similar approach to enable text-only models to perform captioning, reasoning, and question answering that involve videos or images (Yang et al., 2022, 2023; Hu et al., 2022). However, the utility of their approach for multi-modal QG remains largely known.

## 2 The MultiQG-TI Methodology

We now describe the four modules in MultiQG-TI: a question generator module, an image-to-text module, an optical character recognition (OCR) module, and an input formatting module.

**The question generator module.** This module generates the question and is the only trainable module in MultiQG-TI. We adopt a text-based question generator such that its inputs must be all in text format. Adopting a text-based question generator enables us to choose from a wide range of pre-trained text-based generative models, whose training is also often more efficient than their multi-modal counterparts. In this work, we instantiate the question generator with the recent Flan-T5 model (Chung et al., 2022) that have shown to perform strongly on new downstream tasks when fine-tuned on limited task-specific data.

**The image-to-text and OCR modules.** A text-based question generator cannot take any visual input. To solve this problem, we use the image-to-text and OCR modules to interface between the

Table 1: MultiQG-TI (marked bold) significantly outperforms ChatGPT as well as variants with a single modality input across almost all metrics.

Method	BLEU	METEOR	ROUGE	BLEURT
ChatGPT 0 shot	0.014	0.264	0.209	0.448
ChatGPT 1 shot	0.021	0.298	0.208	0.434
ChatGPT 3 shot	0.063	0.332	0.266	0.449
ChatGPT 5 shot	0.088	0.346	0.301	0.464
ChatGPT 7 shot	0.089	0.342	0.307	0.460
<b>MultiQG-TI</b>	<b>0.725</b>	<b>0.829</b>	<b>0.830</b>	<b>0.757</b>
- text only	0.570	0.714	0.718	0.675
- image only	0.714	0.817	0.813	0.760

image and text modalities and extract the visual information from the image format into a textual format appropriate as input for the text-based question generator. In particular, we use the image-to-text module to describe the content in the image in texts, including any objects, scenes, actions, and events. We instantiate this module with the Flan-T5-XXL version of BLIP-2 (Li et al., 2023). While the image-to-text module extracts visually rich signals, it often fails to recognize any text in the image. This is problematic if the majority of the content in the image is text, such as a table. Therefore, we complement the image-to-text module with an OCR module that specializes in extracting the *texts* in the image. We instantiate the OCR module in MultiQG-TI with PaddleOCR (Du et al., 2020).

**The input formatting module.** This module,  $g$ , is a simple function that concatenates the input text and the texts from the input image into one coherent textual input for the question generator model. There are many choices available and one can simply perform a string join operation. In this work, we apply input formatting with the following template: `Generate a question based on the following information. Background: {input_text}. Image: {image_description}. Texts in image: {image_text}.. In this template, {input_text}, {image_description}, and {image_text} are placeholders that will be replaced with the actual input text, the output from the image-to-text model and the output from the OCR module, respectively.`

**Training and inference.** During training, we only update the parameters of the QG module while keeping the other modules fixed. We use the next word prediction as the training objective, which is commonly used in modern language model training (Vaswani et al., 2017). During inference, we proceed as follows: given an input image and text,

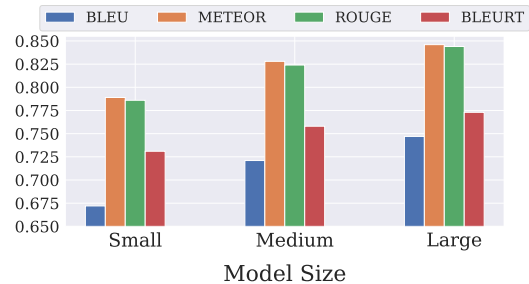


Figure 3: Larger model tends to result in improved QG performance across all metrics.

we first extract the text from the image using image-to-text module and the OCR module, then format them together with the input text, and finally feed the formatted texts to the fine-tuned QG module to generate a question.

### 3 Experiments

**Dataset.** We use the ScienceQA dataset (Lu et al., 2022) throughout our experiments, which we preprocess and split into training, validation, and test splits. All results in this paper are reported on the test split. More details on the dataset and preprocessing steps are in Appendix A.1.

**Baselines.** Because there are no prior work on automatic multi-modal QG, we use off-the-shelf model APIs and variants of MultiQG-TI as the baselines. Specifically, we use **ChatGPT** API (Ouyang et al., 2022) with zero-shot and in-context learning (Kaplan et al., 2020; Wei et al., 2022) with up to seven examples, each of which is formatted exactly the same as our preprocessed data points in the ScienceQA dataset. We also compare with **MultiQG-TI with only a single modality as input** (i.e., either only text or only image).

**evaluation.** We choose four evaluation metrics including **BLEU**, **METEOR**, **ROUGE**, and **BLEURT**, all of which have been widely used in existing QG works. We report all results, except for those using ChatGPT API, based on the average of 4 random, independent runs. More details on the experiment setup, baselines, and evaluation are in Appendices A.2 and A.3.

#### 3.1 Main quantitative results

Table 1 summarizes the main results.<sup>1</sup> These results clearly show that ChatGPT fails at the multi-modal QG task in our setting. Although its performance steadily improves with more examples in the in-context learning setting, ChatGPT trails

<sup>1</sup>For conciseness, we choose not to report standard deviations because all of them are quite small (around 0.002).

Table 2: An example of a question in physics generated by MultiQG-TI.


Input background text	Input image
Magnets can pull or push on other magnets without touching them. When magnets attract, they pull together. When magnets repel, they push apart. These pulls and pushes are called magnetic forces. Magnetic forces are strongest at the magnets’ poles, or ends. Every magnet has two poles: a north pole (N) and a south pole (S). Here are some examples of magnets. Their poles are shown in different colors and labeled. Whether a magnet attracts or repels other magnets depends on the positions of its poles. If opposite poles are closest to each other, the magnets attract. The magnets in the pair below attract. If the same, or like, poles are closest to each other, the magnets repel. The magnets in both pairs below repel.	
MultiQG-TI generated question	
Two magnets are placed as shown. Will these magnets attract or repel each other?	

Table 3: A sufficiently large image-to-text model leads to better QG performance, although the benefit of model size diminishes as the size increases beyond 2.7 billion parameters.

ViT model	bleu_4	meteor	rouge	bleurt
ViT-GPT2 (239M)	0.671	0.79	0.785	0.733
BLIP2-OPT (2.7b)	0.744	0.843	0.843	0.770
BLIP2-OPT (6.7b)	0.743	0.842	0.841	0.773
BLIP2-Flan-T5-XXL (11b)	<b>0.747</b>	<b>0.846</b>	<b>0.844</b>	<b>0.773</b>

MultiQG-TI by a gigantic margin. The comparison between ChatGPT and MultiQG-TI reminds one to be cautious when using ChatGPT in specialized tasks such as multi-modal QG and presents strong empirical evidence that a small, fine-tuned model is still highly relevant in certain generation tasks. Table 1 also demonstrate the benefits of including both the visual and textual information when generating questions because MultiQG-TI outperforms its variants with only textual or only visual input.

### 3.2 Analyses

**The choice of question generators.** We study the impact of the model size of the QG module on the QG performance and summarize the results in Figure 3, where “small”, “medium”, and “large” represent the Flan-T5 variants of 80 million, 250 million, and 780 million parameters, respectively. The figure implies that a larger model generally leads to improved performance across all evaluation metrics. Notably, by fine-tuning only on a few thousand training examples with a modest-sized model, MultiQG-TI achieves high performance,<sup>2</sup> making it appealing for practical use and deployment in resource-constrained settings.

**The choice of image-to-text models.** We also study the impact of the image-to-text models on

<sup>2</sup>As a comparison, some of the latest QG works achieve a BLEURT score of up to 0.67; see the results of a recent QG competition: <https://www.thequestchallenge.org/leaderboard>

the QG performance and summarize the results in Table 3. Specifically, we compare BLIP2-Flan-T5-XXL (11 billion parameters), the image-to-text model we use in MultiQG-TI, to three smaller variants ranging from 239 million to 2.7 billion, and 6.7 billion parameters, respectively. We observe that QG performance improves steadily but minimally after the model becomes larger than 2.7 billion parameters, although the largest model still wins modestly. These results imply that MultiQG-TI may retain the same level of competitiveness even with a smaller off-the-shelf image-to-text model, suggesting more resource-saving opportunities without compromising performance.

**Qualitative examples.** We show an example generated question by MultiQG-TI in Table 2, as well as additional ones in Appendix C. These examples further illustrates MultiQG-TI’s capability in generating fluent, coherent, and meaningful questions from multi-modal scientific contexts. We also provide an in-depth analyses of the errors that MultiQG-TI makes during generation, which we defer to Appendix C due to space constraint.

## 4 Conclusion

We have conducted a first study into automatic multi-modal QG from images and texts. Our proposed solution, MultiQG-TI, is simple, easy-to-use, and highly capable, as evaluated and analyzed on the ScienceQA dataset. Our work opens a myriad of research opportunities. Some of the exciting future directions include: 1) QG with multi-modal inputs and multi-modal outputs; 2) end-to-end vision-language modeling approach for QG; and 3) evaluating and comparing the pedagogical utilities of questions generated from multi-modal sources in real-world educational scenarios.

## References

- Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. 2022. [Flamingo: a visual language model for few-shot learning](#). In *Advances in Neural Information Processing Systems*.
- 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)*.
- Jill Burstein, Geoffrey T LaFlair, Antony John Kunnan, and Alina A von Davier. 2021. A theoretical assessment ecosystem for a digital-first assessment—the duolingo english test. *DRR-21-04*.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. [Scaling instruction-finetuned language models](#).
- Yuning Du, Chenxia Li, Ruoyu Guo, Xiaoting Yin, Weiwei Liu, Jun Zhou, Yifan Bai, Zilin Yu, Yehua Yang, Qingqing Dang, and Haoshuang Wang. 2020. [Pp-ocr: A practical ultra lightweight ocr system](#).
- Xiaodong He and Li Deng. 2017. [Deep learning for image-to-text generation: A technical overview](#). *IEEE Signal Processing Magazine*, 34(6):109–116.
- Michael Heilman and Noah A. Smith. 2010. [Good question! statistical ranking for question generation](#). In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617, Los Angeles, California. Association for Computational Linguistics.
- MD. Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. 2019. [A comprehensive survey of deep learning for image captioning](#). *ACM Computing Surveys*, 51(6):1–36.
- Yushi Hu, Hang Hua, Zhengyuan Yang, Weijia Shi, Noah A. Smith, and Jiebo Luo. 2022. [Promptcap: Prompt-guided task-aware image captioning](#).
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. [Survey of hallucination in natural language generation](#). *ACM Computing Surveys*, 55(12):1–38.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. [Scaling laws for neural language models](#).
- Ravi Kokku, Sharad Sundararajan, Prasenjit Dey, Renuka Sindhgatta, Satya Nitta, and Bikram Sen Gupta. 2018. [Augmenting classrooms with AI for personalized education](#). In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.
- Devang Kulshreshtha, Muhammad Shayan, Robert Belfer, Siva Reddy, Iulian Vlad Serban, and Ekaterina Kochmar. 2022. [Few-shot question generation for personalized feedback in intelligent tutoring systems](#).
- Nguyen-Thanh Le, Tomoko Kojiri, and Niels Pinkwart. 2014. [Automatic question generation for educational applications – the state of art](#). In *Advanced Computational Methods for Knowledge Engineering*, pages 325–338. Springer International Publishing.
- Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. [Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models](#).
- Tianqiao Liu, Qiang Fang, Wenbiao Ding, Hang Li, Zhongqin Wu, and Zitao Liu. 2021. [Mathematical word problem generation from commonsense knowledge graph and equations](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. [Learn to explain: Multimodal reasoning via thought chains for science question answering](#). In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*.
- OpenAI. 2023. [Gpt-4 technical report](#).
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback](#).
- Liangming Pan, Wenqiang Lei, Tat-Seng Chua, and Min-Yen Kan. 2019. [Recent advances in neural question generation](#).

- Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. [Hierarchical text-conditional image generation with clip latents](#).
- Megha Srivastava and Noah Goodman. 2021. [Question generation for adaptive education](#). 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)*. Association for Computational Linguistics.
- Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. [A contrastive framework for neural text generation](#). In *Advances in Neural Information Processing Systems*.
- Alon Talmor, Ori Yoran, Amnon Catav, Dan Lahav, Yizhong Wang, Akari Asai, Gabriel Ilharco, Hananeh Hajishirzi, and Jonathan Berant. 2021. [Multi-modal{qa}: complex question answering over text, tables and images](#). In *International Conference on Learning Representations*.
- 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.
- Zhenhailong Wang, Manling Li, Ruochen Xu, Luwei Zhou, Jie Lei, Xudong Lin, Shuohang Wang, Ziyi Yang, Chenguang Zhu, Derek Hoiem, Shih-Fu Chang, Mohit Bansal, and Heng Ji. 2022. [Language models with image descriptors are strong few-shot video-language learners](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 8483–8497. Curran Associates, Inc.
- 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*. Association for Computational Linguistics.
- Zichao Wang, Andrew S. Lan, Weili Nie, Andrew E. Waters, Phillip J. Grimaldi, and Richard G. Baraniuk. 2018. [QG-net](#). In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*. ACM.
- Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. [Emergent abilities of large language models](#). *Transactions on Machine Learning Research*. Survey Certification.
- Julia White, Amy Burkhardt, Jason Yeatman, and Noah Goodman. 2022. Automated generation of sentence reading fluency test items. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44.
- John H. Wolfe. 1976. [Automatic question generation from text - an aid to independent study](#). In *Proceedings of the ACM SIGCSE-SIGCUE technical symposium on Computer science and education -*. ACM Press.
- Ying Xu, Dakuo Wang, Mo Yu, Daniel Ritchie, Bingsheng Yao, Tongshuang Wu, Zheng Zhang, Toby Li, Nora Bradford, Branda Sun, Tran Hoang, Yisi Sang, Yufang Hou, Xiaojuan Ma, Diyi Yang, Nanyun Peng, Zhou Yu, and Mark Warschauer. 2022. [Fantastic questions and where to find them: FairytaleQA – an authentic dataset for narrative comprehension](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.
- Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. 2022. [Zero-shot video question answering via frozen bidirectional language models](#). In *Advances in Neural Information Processing Systems*.
- Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. 2023. [Mm-react: Prompting chatgpt for multimodal reasoning and action](#).
- Zheng Zhang, Ying Xu, Yanhao Wang, Bingsheng Yao, Daniel Ritchie, Tongshuang Wu, Mo Yu, Dakuo Wang, and Toby Jia-Jun Li. 2022. [StoryBuddy: A human-AI collaborative chatbot for parent-child interactive storytelling with flexible parental involvement](#). In *CHI Conference on Human Factors in Computing Systems*. ACM.
- Zhenjie Zhao, Yufang Hou, Dakuo Wang, Mo Yu, Chengzhong Liu, and Xiaojuan Ma. 2022. [Educational question generation of children storybooks via question type distribution learning and event-centric summarization](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

## A Experiment details

### A.1 Dataset and preprocessing

Each data point in the ScienceQA dataset contains the question text, a background text, and an image. The total number of data points in the ScienceQA dataset is 21,208. We refer readers to [Lu et al. \(2022\)](#) for more details on the dataset. However, the background text and the image are optionally included. As a result, not all data points contain both the background text and the image. We only keep data points that contain all three elements, resulting in 5,942 data points. We further randomly split them into train, validation, and test splits, resulting in 3606/1204/1132 data points in the train/validation/test splits, respectively. For both the remaining texts and images, we did not perform further processing and keep them as-is before feeding them to the MultiQG-TI components that are responsible for processing them.

We note that the MultimodalQA dataset ([Talmor et al., 2021](#)) is also an appropriate dataset choice with rich multi-modal information beyond just texts and images. Because our present work focuses on image and text as input modalities, we leave more complex data modalities for QG for future work.

### A.2 MultiQG-TI model details

**Image-to-text generation.** We use contrastive sampling ([Su et al., 2022](#)) with the following parameters:<sup>3</sup>  $\alpha = 0.6$  and  $k = 4$ , with a temperature of 1, n-gram penalty of 3, and minimum text description length of 30 tokens. For each given image, we sample 10 different text descriptions, rerank them by the image-to-text model’s perplexity, and choose the best description (with the lowest perplexity score) as the final text description for the image, which we will then send to the QG module, together with the OCR module’s output and the input background text.

**QG module training.** We perform all training on a single NVIDIA Quadro RTX 8000 GPU. For all QG module variants that we consider, we use the same training setup. Specifically, we train it with a learning rate of 0.0003 for 8 epochs with early stopping if validation loss does not improve over the most recent 3 epochs. We use a batch size of 3 with a gradient accumulation step of 4, resulting in

<sup>3</sup>See this [blog post](https://huggingface.co/blog/introducing-csearch) for an explanation of the different parameters that appear in contrastive sampling: <https://huggingface.co/blog/introducing-csearch>

an effective batch size of 12 (e.g., the parameters are updated every 12 training steps). We also clip the gradients to 1 to stabilize training. All these training procedures are standard in training text generative models.

**Inference and evaluation.** We use the same contrastive sampling strategy as in image-to-text generation. Additionally, we sample 10 generated questions, rerank them by perplexity, and fetch the best-ranked sample as the final generated question for each input text-image pair in the test set. All evaluations are conducted on this “top-1” setting. For each individual run, we perform the above sampling strategy with a different seed to obtain a different set of generated questions for each input in the test set. We then perform the same evaluation on each generated set and then average the results, resulting in the averaged quantitative evaluations reported in the main paper.

**Remarks.** MultiQG-TI leverages readily available, open-source tools to solve the new problem of multi-modal question generation. Its modular design makes it flexible and easily adaptable, enabling one to upgrade a component when a more capable one becomes available. Moreover, the only trainable component is the question generator. There are many choices available for this component, any of which can achieve competitive performance with relatively limited model sizes, making it suitable for low-resource training settings. An end-to-end multi-modal QG model is still methodologically interesting and we leave this as a future work.

### A.3 ChatGPT baseline

We use the `gpt-3.5-turbo-0301` model API throughout our experiments. The system message we give to the model at the beginning of the API call is as follows: You are a helpful assistant. Your job is to generate a question, which consists of a question background/context and the question itself, given the user’s provided context information, which consists of an instruction, background, subject, topic, and category. Your answer should be in the following template: ‘Question context: ... Question: ...’. After that, for zero-shot QG, we send the



templated input background text, OCR extracted text from the input image, and the text description of the input image to the API, formatted exactly as what we would do for MultiQG-TI. For few-shot QG, we construct each example as a pair of input and output, where the input is the templated input consisting of the input text and texts extracted from the input image, and the output is the corresponding question text to the input text and image. We only perform generation once for each setting and for each input to avoid incurring higher costs of making OpenAI API calls.

**Selecting examples for in-context learning.** We perform a basic cosine similarity search for each input context and image pairs. Specifically, we first encode each formatted input text (recall, it contains the input background text, the image description, and the texts in the image) as a vector using the SentenceTransformers<sup>4</sup>. Then, for each formatted input in the test set, we perform a similarity search, computing its cosine similarity with every formatted input in the training set, and select up to seven most similar formatted input as the examples to be used in prompting ChatGPT in the few-shot in-context learning setting.

## B Additional literature review

The MultimodalQA dataset (Talmor et al., 2021) actually involves a cursory description of generating questions from multiple sources. However, the QG process described therein relies on human annotation, a manual process that cannot achieve automatic QG and therefore is neither a baseline to our work nor related to our goal of automatic QG.

Recent research has demonstrated the impressive capabilities of models that can connect data from multiple modalities, such as generating images from texts (Ramesh et al., 2022) and vice versa (He and Deng, 2017). Specifically related to our work, recent advances in vision-language models (Alayrac et al., 2022; Li et al., 2023; OpenAI, 2023) enable models to converse with a user given both texts and images. However, most demonstrated use cases of these models are in casual dialogues (Li et al., 2023), image captioning (Hossain et al., 2019), and visual question answering (Antol et al., 2015). The utilities of these models for QG remain largely unknown.

---

<sup>4</sup><https://www.sbert.net/>

## C Additional results

### Additional examples of generated questions.

We provide additional generation examples in Table 4 for chemistry, physics, and biology, respectively. These examples corroborate with the one in the main text and demonstrate the capability of MultiQG-TI in generating reasonable questions from image and text inputs.

### Qualitative generation error analysis.

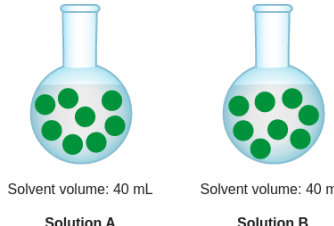
MultiQG-TI is not without problems. In Table 5, we provide an exemplary erroneous generated question to illustrate the typical problems that MultiQG-TI has when performing QG.

In our observation, there are two major sources of error. The first one comes from the mistakes cascaded from the image-to-text model. In the example in Table 5, the object in the image is dolerite, but the image-to-text model in MultiQG-TI recognizes it as granite, resulting in the image description “a black piece of granite on a white background”. As a result, the question generator, which generates the question conditioned on the image description, picks up the wrongly reconigized object “granite” and use it to generate a question on granite instead of on dolerite.

The second source of error comes from hallucination, a major bottleneck preventing language models from real-world, high-stake use scenarios (Ji et al., 2023). MultiQG-TI is not immune to this problem. In the example in Table 5, the question generator produces the phrase “pure substance”, which is neither a property of dolerite nor granite because both are mixtures.

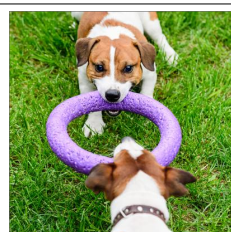
These are challenging issues to tackle. For example, it is even difficult for a non-expert to identify the object in the image in Table 5. Similarly, it is difficult to verify the factual correctness of the generated question without resorting to external sources such as web search and textbooks. Reducing these errors would require improvements to the image-to-text model and mitigating hallucination in language models, both of which remain active areas of research.

Table 4: Additional examples of science questions generated by MultiQG-TI.

Input background text	Input image
<p>A solution is made up of two or more substances that are completely mixed. In a solution, solute particles are mixed into a solvent. The solute cannot be separated from the solvent by a filter. For example, if you stir a spoonful of salt into a cup of water, the salt will mix into the water to make a saltwater solution. In this case, the salt is the solute. The water is the solvent. The concentration of a solute in a solution is a measure of the ratio of solute to solvent. Concentration can be described in terms of particles of solute per volume of solvent. <math>\text{concentration} = \text{particles of solute} / \text{volume of solvent}</math></p>	 <p>Solvent volume: 40 mL      Solvent volume: 40 mL</p> <p>Solution A                      Solution B</p>


**MultiQG-TI generated question**

The diagram below is a model of two solutions. Each pink ball represents one particle of solute. Which solution has a higher concentration of pink particles?

Input background text	Input image
<p>A force is a push or a pull that acts on an object. Each force acts on an object in a certain direction. If two forces act on an object in opposite directions, they are called opposing forces.</p>	

**MultiQG-TI generated question**


The text below describes a pair of opposing forces. Opposing forces act on an object in opposite directions. Read the text. Then answer the question below. Two puppies, Rusty and Coco, play with a toy. Think about two of the forces that act on the toy: Coco pulls toward herself. Rusty pulls away from Coco. As Coco pulls on the toy, what is the direction of the opposing force?

Input background text	Input image
<p>Offspring genotypes: homozygous or heterozygous? How do you determine whether an organism is homozygous or heterozygous for a gene? Look at the alleles in the organism's genotype for that gene. An organism with two identical alleles for a gene is homozygous for that gene. If both alleles are dominant, the organism is homozygous dominant for the gene. If both alleles are recessive, the organism is homozygous recessive for the gene. An organism with two different alleles for a gene is heterozygous for that gene. In a Punnett square, each box represents a different outcome, or result. Each of the four outcomes is equally likely to happen. Each box represents one way the parents' alleles can combine to form an offspring's genotype. Because there are four boxes in the Punnett square, there are four possible outcomes. An event is a set of one or more outcomes. The probability of an event is a measure of how likely the event is to happen. .... (truncated due to excessive length)</p>	

**MultiQG-TI generated question**

In a group of rainbow trout, some individuals have a greenish-brown body and others have a blue body. In this group, the gene for the body color trait has two alleles. The allele for a blue body (e) is recessive to the allele for a greenish-brown body (E). This Punnett square shows a cross between two rainbow trout. What is the probability that a rainbow trout produced by this cross will be heterozygous for the body color gene?

Table 5: An illustrative erroneous question generated by MultiQG-TI. The texts marked in red are the erroneous parts. The word “granite” results from the image-to-text model, which incorrectly recognizes the dolerite in the image as granite. The phrase “pure substance” is neither a property of dolerite nor granite.

Input background text	Input image
<p>Minerals are the building blocks of rocks. A rock can be made of one or more minerals. Minerals and rocks have the following properties: Property   Mineral   Rock It is a solid.   Yes   Yes It is formed in nature.   Yes   Yes It is not made by organisms.   Yes   Yes It is a pure substance.   Yes   No It has a fixed crystal structure.   Yes   No You can use these properties to tell whether a substance is a mineral, a rock, or neither. Look closely at the last three properties: Minerals and rocks are not made by organisms. Organisms make their own body parts. For example, snails and clams make their shells. Because they are made by organisms, body parts cannot be minerals or rocks. Humans are organisms too. So, substances that humans make by hand or in factories are not minerals or rocks. A mineral is a pure substance, but a rock is not. A pure substance is made of only one type of matter. Minerals are pure substances, but rocks are not. Instead, all rocks are mixtures. A mineral has a fixed crystal structure, but a rock does not. The crystal structure of a substance tells you how the atoms or molecules in the substance are arranged. Different types of minerals have different crystal structures, but all minerals have a fixed crystal structure. This means that the atoms and molecules in different pieces of the same type of mineral are always arranged the same way. However, rocks do not have a fixed crystal structure. So, the arrangement of atoms or molecules in different pieces of the same type of rock may be different!</p>	
<b>MultiQG-TI generated question</b>	
<p><b>Granite</b> has the following properties: <b>pure substance</b> solid fixed crystal structure naturally occurring not made by living things Question: Is granite a mineral or a rock?</p>	

# Inspecting Spoken Language Understanding from Kids for Basic Math Learning at Home

**Eda Okur**

Intel Labs, USA  
eda.okur@intel.com

**Saurav Sahay**

Intel Labs, USA  
saurav.sahay@intel.com

**Roddy Fuentes Alba**

Intel Labs, Mexico  
roddy.fuentes.alba@intel.com

**Lama Nachman**

Intel Labs, USA  
lama.nachman@intel.com

## Abstract

Enriching the quality of early childhood education with interactive math learning at home systems, empowered by recent advances in conversational AI technologies, is slowly becoming a reality. With this motivation, we implement a multimodal dialogue system to support play-based learning experiences at home, guiding kids to master basic math concepts. This work explores Spoken Language Understanding (SLU) pipeline within a task-oriented dialogue system developed for Kid Space, with cascading Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU) components evaluated on our home deployment data with kids going through gamified math learning activities. We validate the advantages of a multi-task architecture for NLU and experiment with a diverse set of pretrained language representations for Intent Recognition and Entity Extraction tasks in the math learning domain. To recognize kids' speech in realistic home environments, we investigate several ASR systems, including the commercial Google Cloud and the latest open-source Whisper solutions with varying model sizes. We evaluate the SLU pipeline by testing our best-performing NLU models on noisy ASR output to inspect the challenges of understanding children for math learning in authentic homes.

## 1 Introduction and Background

The ongoing progress in Artificial Intelligence (AI) based advanced technologies can assist humanity in reducing the most critical inequities around the globe. The recent widespread interest in conversational AI applications presents exciting opportunities to showcase the positive societal impact of these technologies. The language-based AI systems have already started to mature to a level where we may soon observe their influences in mitigating the most pressing global challenges. Education is among the top priority improvement areas identified by the United Nations (UN) (i.e., poverty,

hunger, healthcare, and education). In particular, increasing the inclusiveness and quality of education is within the UN development goals<sup>1</sup> with utmost urgency. One of the preeminent ways to diminish societal inequity is promoting STEM (i.e., Science, Technology, Engineering, Math) education, specifically ensuring that children succeed in mathematics. It is well-known that acquiring basic math skills at younger ages builds students up for success, regardless of their future career choices (Cesarone, 2008; Torpey, 2012). For math education, interactive learning environments through gamification present substantial leverages over more traditional learning settings for studying elementary math subjects, particularly with younger learners (Skene et al., 2022). With that goal, conversational AI technologies can facilitate this interactive learning environment where students can master fundamental math concepts. Despite these motivations, studying spoken language technologies for younger kids to learn basic math is a vastly uncharted area of AI.

This work discusses a modular goal-oriented Spoken Dialogue System (SDS) specifically targeted for kids to learn and practice basic math concepts at home setup. Initially, a multimodal dialogue system (Sahay et al., 2019) is implemented for Kid Space (Anderson et al., 2018), a gamified math learning application for deployment in authentic classrooms. During this preliminary real-world deployment at an elementary school, the COVID-19 pandemic impacted the globe, and school closures forced students to switch to online learning options at home. To support this sudden paradigm shift to at-home learning, previous school use cases are redesigned for new home usages, and our dialogue system is recreated to deal with interactive math games at home. While the play-based learning activities are adjusted for home usages with a much simpler setup, the multimodal aspects of these games are partially preserved along with the

<sup>1</sup><https://sdgs.un.org/goals>

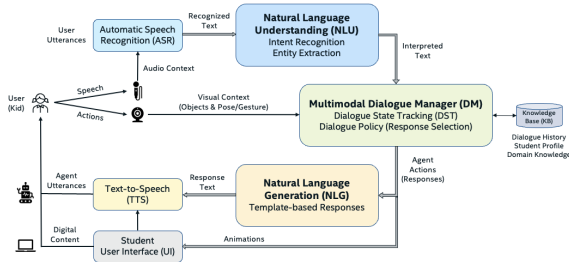


Figure 1: Multimodal Dialogue System Pipeline

fundamental math concepts for early childhood education. These math skills cover using ones and tens to construct numbers and foundational arithmetic concepts and operations such as counting, addition, and subtraction. The multimodal aspects of these learning games include kids’ spoken interactions with the system while answering math questions and carrying out game-related conversations, physical interactions with the objects (i.e., placing cubes and sticks as manipulatives) on a visually observed playmat, performing specific pose and gesture-based actions as part of these interactive games (e.g., jumping, standing still, air high-five).

Our domain-specific SDS pipeline (see Figure 1) consists of multiple cascaded components, namely Automatic Speech Recognition (ASR), Natural Language Understanding (NLU), Multimodal Dialogue Manager (DM), Natural Language Generation (NLG), and Text-to-Speech (TTS) synchronizing the agent utterances with virtual character animations on Student User Interface (UI). Here we concentrate on the Spoken Language Understanding (SLU) task on kids’ speech at home environments while playing basic math games. Such application-dependent SLU approaches commonly involve two main modules applied sequentially: (i) Speech-to-Text (STT) or ASR module that recognizes speech and transcribes the spoken utterances into text, and (ii) NLU module that interprets the semantics of those utterances by processing the transcribed text. NLU is one of the most integral components of these goal-oriented dialogue systems. It empowers user-agent interactions by understanding the meaning of user utterances via performing domain-specific sub-tasks. Intent Recognition (IR) and Named Entity Recognition (NER) are essential sub-tasks within the NLU module to resolve the complexities of human language and extract meaningful information for the application at hand. Given a user utterance as input, the Intent Classification aims to identify the user’s intention (i.e.,

what the user desires to achieve with that interaction) and categorize the user’s objective at that conversational turn. The Entity Extraction targets locating and classifying entities (i.e., specific terms representing existing things such as person names, locations, and organizations) mentioned in user utterances into predefined task-specific categories.

In this study, we present our efforts to convert the task-oriented SDS (Okur et al., 2022b) designed for school use cases (Aslan et al., 2022) to home usages after COVID-19 and inspect the performance of individual SDS modules evaluated on the home deployment data we recently collected from 12 kids individually at their homes. The current work focuses on assessing and improving the SLU task performance on kids’ utterances at home by utilizing this real-world deployment data. We first investigate the ASR and NLU module evaluations independently. Then, we inspect the overall SLU pipeline (ASR+NLU) performance on kids’ speech by evaluating our NLU tasks on ASR output (i.e., recognized text) at home environments. As the erroneous and noisy speech recognition output would lead to incorrect intent and entity predictions, we aim to understand these error propagation consequences with SLU for children in the math learning domain. We experiment with various recent ASR solutions and diverse model sizes to gain more insights into their capabilities to recognize kids’ speech at home. We then analyze the effects of these ASR engines on understanding intents and extracting entities from children’s utterances. We discuss our findings and observations for potential enhancements in future deployments of this multimodal dialogue system for math learning at home.

## 2 Related Work

### 2.1 Conversational AI for Math Learning

With the ultimate goal of improving the quality of education, there has been a growing enthusiasm for exploiting AI-based intelligent systems to boost students’ learning experiences (Chassignol et al., 2018; Aslan et al., 2019; Jia et al., 2020; Zhai et al., 2021; Baker, 2021). Among these, interactive frameworks that support guided play-based learning spaces revealed significant advantages for math learning (Pires et al., 2019; Sun et al., 2021; Richey et al., 2021), especially for building foundational math skills in early childhood education (Nrupatunga et al., 2021; Skene et al., 2022). To attain this level of interactivity within smart

learning spaces, developing innovative educational applications by utilizing language-based AI technologies is in growing demand (Taghipour and Ng, 2016; Lende and Raghuvanshi, 2016; Raamadurai et al., 2019; Cahill et al., 2020; Chan et al., 2021; Rathod et al., 2022). In particular, designing conversational agents for intelligent tutoring is a compelling yet challenging area of research, with several attempts presented so far (Winkler and Söllner, 2018; Wambsganss et al., 2020; Winkler et al., 2020; Datta et al., 2020; Okonkwo and Ade-Ibijola, 2021; Wollny et al., 2021), most of them focusing on language learning (Bibauw et al., 2022; Tyen et al., 2022; Zhang et al., 2022).

In the math education context, earlier conversational math tutoring applications exist, such as SKOPE-IT (Nye et al., 2018), which is based on AutoTutor (Graesser et al., 2005) and ALEKS (Falmagne et al., 2013), and MathBot (Grossman et al., 2019). These are often text-based online systems following strict rules in conversational graphs. Later, various studies emerged at the intersection of cutting-edge AI techniques and math learning (Mansouri et al., 2019; Huang et al., 2021; Azerbayev et al., 2022; Uesato et al., 2022; Yang et al., 2022). Among those, employing advanced language understanding methods to assist math learning is relatively new (Peng et al., 2021; Shen et al., 2021; Loginova and Benoit, 2022; Reusch et al., 2022). The majority of those recent work leans on exploring language representations for math-related tasks such as mathematical reasoning, formula understanding, math word problem-solving, knowledge tracing, and auto-grading, to name a few. Recently, TalkMoves dataset (Suresh et al., 2022a) was released with K-12 math lesson transcripts annotated for discursive moves and dialogue acts to classify teacher talk moves in math classrooms (Suresh et al., 2022b).

For the conversational AI tasks, the latest large language models (LLMs) based chatbots, such as BlenderBot (Shuster et al., 2022) and ChatGPT (OpenAI, 2022), gained a lot of traction in the education community (Tack and Piech, 2022; Kasneci et al., 2023), along with some concerns about using generative models in tutoring (Macina et al., 2023; Cotton et al., 2023). ChatGPT is a general-purpose open-ended interaction agent trained on internet-scale data. It is an end-to-end dialogue model without explicit NLU/Intent Recognizer or DM, which currently cannot fully comprehend the

multimodal context and proactively generate responses to nudge children in a guided manner without distractions. Using these recent chatbots for math learning is still in the early stages because they are known to miss basic mathematical abilities and carry reasoning flaws (Frieder et al., 2023), revealing a lack of common sense. Moreover, they are known to be susceptible to triggering inappropriate or harmful responses and potentially perpetuate human biases since they are trained on internet-scale data and require carefully-thought guardrails.

On the contrary, our unique application is a task-oriented math learning spoken dialogue system designed to perform learning activities, following structured educational games to assist kids in practicing basic math concepts at home. Our SDS does not require massive amounts of data to understand kids and generate appropriate adaptive responses, and the lightweight models can run locally on client machines. In addition, our solution is multimodal, intermixing the physical and digital hybrid learning experience with audio-visual understanding, object recognition, segmentation, tracking, and pose and gesture recognition.

## 2.2 Spoken Language Understanding

Conventional pipeline-based dialogue systems with supervised learning are broadly favored when initial domain-specific training data is scarce to bootstrap the task-oriented SDS for future data collection (Serban et al., 2018; Budzianowski et al., 2018; Mehri et al., 2020). Deep learning-based modular dialogue frameworks and practical toolkits are prominent in academic and industrial settings (Bocklisch et al., 2017; Burtsev et al., 2018; Reyes et al., 2019). For task-specific applications with limited in-domain data, current SLU systems often use a cascade of two neural modules: (i) ASR maps the input audio to text (i.e., transcript), and (ii) NLU predicts intent and slots/entities from this transcript. Since our main focus in this work is investigating the SLU pipeline, we briefly summarize the existing NLU and ASR solutions.

### 2.2.1 Language Representations for NLU

The NLU component processes input text, often detects intents, and extracts referred entities from user utterances. For the mainstream NLU tasks of Intent Classification and Entity Recognition, jointly trained multi-task models are proposed (Liu and Lane, 2016; Zhang and Wang, 2016; Goo et al., 2018) with hierarchical learning approaches (Wen

et al., 2018; Okur et al., 2019; Vanzo et al., 2019). Transformer architecture (Vaswani et al., 2017) is a game-changer for several downstream language tasks. With Transformers, BERT (Devlin et al., 2019) is presented, which became one of the most pivotal breakthroughs in language representations, achieving high performance in various tasks, including NLU. Later, Dual Intent and Entity Transformer (DIET) architecture (Bunk et al., 2020) is invented as a lightweight multi-task NLU model. On multi-domain NLU-Benchmark data (Liu et al., 2021b), the DIET model outperformed fine-tuning BERT for joint Intent and Entity Recognition.

For BERT-based autoencoding approaches, RoBERTa (Liu et al., 2019) is presented as a robustly optimized BERT model for sequence and token classification. The Hugging Face introduced a smaller, lighter general-purpose language representation model called DistilBERT (Sanh et al., 2019) as the knowledge-distilled version of BERT. ConveRT (Henderson et al., 2020) is proposed as an efficiently compact model to obtain pretrained sentence embeddings as conversational representations for dialogue-specific tasks. LaBSE (Feng et al., 2022) is a pretrained multilingual model producing language-agnostic BERT sentence embeddings that achieve promising results in text classification.

The GPT family of autoregressive LLMs, such as GPT-2 (Radford et al., 2019) and GPT-3 (Brown et al., 2020), perform well at what they are pretrained for, i.e., text generation. GPT models can also be adopted for NLU, supporting few-shot learning capabilities, and NLG in task-oriented dialogue systems (Madotto et al., 2020; Liu et al., 2021a). XLNet (Yang et al., 2019) applies autoregressive pretraining for representation learning that adopts Transformer-XL (Dai et al., 2019) as a backbone model and works well for language tasks with lengthy contexts. DialoGPT (Zhang et al., 2020) extends GPT-2 as a large-scale neural response generation model for multi-turn conversations trained on Reddit discussions, whose representations can be exploited in dialogue tasks.

For language representations to be utilized in math-related tasks, MathBERT (Shen et al., 2021) is introduced as a math-specific BERT model pretrained on large math corpora. Later, Math-aware-BERT and Math-aware-RoBERTa models (Reusch et al., 2022) are proposed based on BERT and RoBERTa, pretrained on Math Stack Exchange<sup>2</sup>.

<sup>2</sup><https://math.stackexchange.com/>

## 2.2.2 Speech Recognition with Kids

Speech recognition technology has been around for some time, and numerous ASR solutions are available today, both commercial and open-source. Rockhopper ASR (Stemmer et al., 2017) is an earlier low-power speech recognition engine with LSTM-based language models, where its acoustic models are trained using an open-source Kaldi speech recognition toolkit (Povey et al., 2011). Google Cloud Speech-to-Text<sup>3</sup> is a prominent commercial ASR service powered by advanced neural models and designed for speech-dependant applications. Until recently, Google STT API was arguably the leader in ASR services for recognition performance and language coverage. Franck Deroncourt (2018) reported that Google ASR could reach a word error rate (WER) of 12.1% on LibriSpeech clean dataset (28.8% on LibriSpeech other) (Panayotov et al., 2015) at that time, which is improved drastically over time. Recently, Open AI released Whisper ASR (Radford et al., 2022) as a game-changer speech recognizer. Whisper models are pretrained on a vast amount of labeled audio-transcription data (i.e., 680k hours), unlike its predecessors (e.g., Wav2Vec 2.0 (Baevski et al., 2020) is trained on 60k hours of unlabeled audio). 117k hours of this data are multilingual, which makes Whisper applicable to over 96 languages, including low-resourced ones. Whisper architecture follows a standard Transformer-based encoder-decoder as many speech-related models (Latif et al., 2023). The Whisper-base model is reported to achieve 5.0% & 12.4% WER on LibriSpeech clean & other.

Although speech recognition systems are substantially improving to achieve human recognition levels, problems still occur, especially in noisy environments, with users having accents and dialects or underrepresented groups like kids. Child speech brings distinct challenges to ASR (Stemmer et al., 2003; Gerosa et al., 2007; Yeung and Alwan, 2018), such as data scarcity and highly varied acoustic, linguistic, physiological, developmental, and articulatory characteristics compared to adult speech (Claus et al., 2013; Shivakumar and Georgiou, 2020; Bhardwaj et al., 2022). Thus, WER for children's voices is reported two-to-five times worse than for adults (Wu et al., 2019), as the younger the child, the poorer ASR performs. There exist efforts to mitigate these difficulties of speech recognition with kids (Shivakumar et al.,

<sup>3</sup><https://cloud.google.com/speech-to-text/>

2014; Duan and Chen, 2020; Booth et al., 2020; Kelly et al., 2020; Rumberg et al., 2021; Yeung et al., 2021). Few studies also focus on speech technologies in educational settings (Reeder et al., 2015; Blanchard et al., 2015; Bai et al., 2021, 2022; Dutta et al., 2022), often for language acquisition, reading comprehension, and story-telling activities.

### 3 Methods

#### 3.1 Home Learning Data and Use Cases

We utilize two datasets for gamified basic math learning at home usages. The first set is a proof-of-concept (POC) data manually constructed based on User Experience (UX) studies (e.g., detailed scripts for new home use cases) and partially adopted from our previous school data (Okur et al., 2022a). This POC data is used to train and cross-validate various NLU models to develop the best practices in later home deployments. The second set is our recent home deployment data collected from 12 kids (ages 7-8) experiencing our multimodal math learning system at authentic homes. The audio-visual data is transcribed manually, and user utterances in these reference transcripts are annotated for intent and entity types we identified for each learning activity at home. Table 1 compares the NLU statistics for Kid Space Home POC and Deployment datasets. Manually transcribed children’s utterances in deployment data are employed to test our best NLU models trained on POC data. We run multiple ASR engines on audio recordings from home deployment data, where automatic transcripts (i.e., ASR output) are utilized to compute WER to assess ASR model performances on kids’ speech. We also evaluate the SLU pipeline (ASR+NLU) by testing NLU models on ASR output from deployment data.

The simplified home deployment setup includes a playmat with physical manipulatives, a laptop with a built-in camera, a wireless lavalier mic, and a depth camera on a tripod. Home use cases follow a particular flow of activities designed for play-based learning in early childhood education. These activities are Introduction (Meet & Greet), Warm-up Game (Red Light Green Light), Training Game, Learning Game, and Closure (Dance Party). After meeting with the virtual character and playing jumping games, the child starts the training game, where the agent asks for help planting flowers. The agent presents tangible manipulatives, cubes representing ones and sticks representing tens, and instructs the kid to answer ba-

NLU Data Statistics	POC	Deployment
# Intents Types	13	12
Total # Utterances	4091	733
# Entity Types	3	3
Total # Entities	2244	497
Min # Utterances per Intent	105	1
Max # Utterances per Intent	830	270
Avg # Utterances per Intent	314.7	61.1
Min # Tokens per Utterance	1	1
Max # Tokens per Utterance	40	33
Avg # Tokens per Utterance	4.49	2.30
# Unique Tokens (Vocab Size)	702	149
Total # Tokens	18364	1689

Table 1: Kid Space Home POC and Deployment Data

sic math questions and construct numbers using these objects, going through multiple rounds of practice questions where flowers in child-selected colors bloom as rewards. In the actual learning game, the agent presents clusters of questions involving ones & tens, and the child provides verbal (e.g., stating the numbers) and visual answers (e.g., placing the cubes and sticks on the playmat, detected by the overhead camera). The agent provides scaffolding utterances and performs animations to show and tell how to solve basic math questions. The interaction ends with a dance party to celebrate achievements and say goodbyes in closure. Some of our intents can be considered generic (e.g., *state-name*, *affirm*, *deny*, *repeat*, *out-of-scope*), but some are highly domain-specific (e.g., *answer-flowers*, *answer-valid*, *answer-others*, *state-color*, *had-fun-a-lot*, *end-game*) or math-related (e.g., *state-number*, *still-counting*). The entities we extract are activity-specific (i.e., *name*, *color*) and math-related (i.e., *number*).

#### 3.2 NLU and ASR Models

Customizing open-source Rasa framework (Bocklisch et al., 2017) as a backbone, we investigate several NLU models for Intent Recognition and Entity Extraction tasks to implement our math learning conversational AI system for home usage. Our baseline approach is inspired by the StarSpace (Wu et al., 2018) method, a supervised embedding-based model maximizing the similarity between utterances and intents in shared vector space. We enrich this simple text classifier by incorporating SpaCy (Honnibal et al., 2020) pre-



trained language models<sup>4</sup> for word embeddings as additional features in the NLU pipeline. CRF Entity Extractor (Lafferty et al., 2001) with BILOU tagging is also part of this baseline NLU. For home usages, we explore the advantages of switching to a more recent DIET model<sup>5</sup> for joint Intent and Entity Recognition, a multi-task architecture with two-layer Transformers shared for NLU tasks. DIET leverages combining dense features (e.g., any given pretrained embeddings) with sparse features (e.g., token-level encodings of char n-grams). To observe the net benefits of DIET, we first pass the identical SpaCy embeddings used in our baseline (StarSpace) as dense features to DIET. Then, we adopt DIET with pretrained BERT<sup>6</sup>, RoBERTa<sup>7</sup>, and DistilBERT<sup>8</sup> word embeddings, as well as ConveRT<sup>9</sup> and LaBSE<sup>10</sup> sentence embeddings to inspect the effects of these autoencoding-based language representations on NLU performance (see 2.2.1 for more details). We also evaluate pretrained embeddings from models using autoregressive training such as XLNet<sup>11</sup>, GPT-2<sup>12,13</sup>, and DialoGPT<sup>14</sup> on top of DIET. Next, we explore recently-proposed math-language representations pretrained on math data for our basic math learning dialogue system. MathBERT (Shen et al., 2021) is pretrained on large math corpora (e.g., curriculum, textbooks, MOOCs, arXiv papers) covering pre-k to college-graduate materials. We enhance DIET by incorporating embeddings from MathBERT-base<sup>15</sup> and MathBERT-custom<sup>16</sup> models, pretrained with BERT-base original and math-customized vocabularies, respectively. Math-aware-BERT<sup>17</sup> and Math-aware-RoBERTa<sup>18</sup> mod-

<sup>4</sup>[https://github.com/explosion/spacy-models/releases/tag/en\\_core\\_web\\_md-3.5.0](https://github.com/explosion/spacy-models/releases/tag/en_core_web_md-3.5.0)

<sup>5</sup>Please check Bunk et al. (2020) for hyper-parameter tuning, hardware specs, and computational costs.

<sup>6</sup><https://huggingface.co/bert-base-uncased>

<sup>7</sup><https://huggingface.co/roberta-base>

<sup>8</sup><https://huggingface.co/distilbert-base-uncased>

<sup>9</sup><https://github.com/connorbrinton/polyai-models/releases>

<sup>10</sup><https://huggingface.co/rasa/LaBSE>

<sup>11</sup><https://huggingface.co/xlnet-base-cased>

<sup>12</sup><https://huggingface.co/gpt2>

<sup>13</sup>Excluded GPT-3 and beyond that are not open-source.

<sup>14</sup><https://huggingface.co/microsoft/DialoGPT-medium>

<sup>15</sup><https://huggingface.co/tbs17/MathBERT>

<sup>16</sup><https://huggingface.co/tbs17/MathBERT-custom>

<sup>17</sup>[https://huggingface.co/AnReu/math\\_pretrained\\_bert](https://huggingface.co/AnReu/math_pretrained_bert)

<sup>18</sup>[https://huggingface.co/AnReu/math\\_pretrained\\_roberta](https://huggingface.co/AnReu/math_pretrained_roberta)

els (Reusch et al., 2022) are initialized from BERT-base and RoBERTa-base, and further pretrained on Math StackExchange<sup>19</sup> with extra LaTeX tokens to better tokenize math formulas for ARQMath-3 tasks (Mansouri et al., 2022). We exploit these representations with DIET to investigate their effects on our NLU tasks in the basic math domain.

For the ASR module, we explore three main speech recognizers for our math learning application at home, which are explained further in 2.2.2. Rockhopper ASR<sup>20</sup> is the baseline local approach previously inspected, which can be adjusted slightly for kids. Its acoustic models rely on Kaldi<sup>21</sup> generated resources and are trained on default adult speech data. In the past explorations, when Rockhopper’s language models fine-tuned with limited in-domain kids’ utterances (Sahay et al., 2021) from previous school usages, WER decreased by 40% for kids but remained 50% higher than adult WER. Although this small-scale baseline solution is unexpected to reach Google Cloud ASR performance, Rockhopper has a few other advantages for our application since it can run offline locally on low-power devices, which could be better for security, privacy, latency, and cost (relative to cloud-based ASR services). Google ASR is a commercial cloud solution providing high-quality speech recognition service but requiring connectivity and payment, which cannot be adapted or fine-tuned as Rockhopper. The third ASR approach we investigate is Whisper<sup>22</sup>, which combines the best of both worlds as it is an open-source adjustable solution that can run locally, achieving new state-of-the-art (SOTA) results. We inspect three configurations of varying model sizes (i.e., base, small, and medium) to evaluate the Whisper ASR for our home math learning usage with kids.

## 4 Experimental Results

To build the NLU module of our SLU pipeline, we train Intent and Entity Classification models and cross-validate them over the Kid Space Home POC dataset to decide upon the best-performing NLU architectures moving forward for home. Table 2 summarizes the results of model selection experiments with various NLU models. We report the average of 5 runs, and each run involves a 10-fold

<sup>19</sup><https://archive.org/download/stackexchange>

<sup>20</sup>[https://docs.openvino.ai/2018\\_R5/\\_samples\\_speech\\_sample\\_README.html](https://docs.openvino.ai/2018_R5/_samples_speech_sample_README.html)

<sup>21</sup><https://github.com/kaldi-asr/kaldi>

<sup>22</sup><https://github.com/openai/whisper>

NLU Model	Intent Detection	Entity Extraction
StarSpace+SpaCy	92.71±0.25	97.08±0.21
DIET+SpaCy	94.29±0.05	98.38±0.12
DIET+BERT	97.25±0.23	99.23±0.02
DIET+RoBERTa	95.50±0.18	99.11±0.12
DIET+DistilBERT	97.41±0.20	99.49±0.12
DIET+ConveRT	<b>98.80±0.25</b>	99.61±0.03
DIET+LaBSE	98.19±0.18	<b>99.72±0.04</b>
DIET+XLNet	94.99±0.19	98.38±0.14
DIET+GPT-2	95.35±0.27	99.01±0.27
DIET+DialogPT	96.00±0.49	98.94±0.12
DIET+MathBERT-base	94.55±0.22	98.10±0.21
DIET+MathBERT-custom	94.61±0.34	97.48±0.29
DIET+Math-aware-BERT	95.95±0.15	98.94±0.19
DIET+Math-aware-RoBERTa	94.20±0.16	98.75±0.21

Table 2: NLU Model Selection Results in F1-scores (%) Evaluated on Kid Space Home POC Data (10-fold CV)

cross-validation (CV) on POC data. Compared to the baseline StarSpace algorithm, we gain almost 2% F1 score for intents and more than 1% F1 for entities with multi-task DIET architecture. For language representations, we observe that incorporating DIET with the BERT family of embeddings from autoencoders achieves higher F1 scores relative to the GPT family of embeddings from autoregressive models. We cannot reveal any benefits of employing math-specific representations with DIET, as all such models achieve worse than DIET+BERT results. One reason we identify is the mismatch between our early math domain and advanced math corpora, including college-level math symbols and equations, that these models trained on. Another reason could be that such embeddings are pretrained on smaller math corpora (e.g., 100 million tokens) compared to massive-scale generic corpora (e.g., 3.3 billion words) that BERT models use for training. DIET+ConveRT is the clear winner for intents and achieves second-best but very close results for entities compared to DIET+LaBSE. ConveRT and LaBSE are both sentence-level embeddings, but ConveRT performs well on dialogue tasks as it is pretrained on large conversational corpora, including Reddit discussions. Based on these results, we select DIET+ConveRT as the final multi-task architecture for our NLU tasks at home.

Next, we evaluate our NLU module on Kid Space Home Deployment data collected at authentic homes over 12 sessions with 12 kids. Each child goes through 5 activities within a session, as described in 3.1. In Table 3, we observe overall F1% drops ( $\Delta$ ) of 4.6 for intents and 0.3 for entities when our best-performing DIET+ConveRT models are tested on home deployment data. These findings are expected and relatively lower than

Activity	Intent Detection			Entity Extraction		
	POC	Deploy	$\Delta$	POC	Deploy	$\Delta$
Intro (Meet & Greet)	99.9	97.3	-2.6	99.2	97.4	-1.8
Warm-up Game	98.8	93.4	-5.4	-	-	-
Training Game	98.4	94.2	-4.2	99.9	99.8	-0.1
Learning Game	98.9	94.3	-4.6	99.8	99.4	-0.4
Closure (Dance)	98.8	98.7	-0.1	-	-	-
<b>All Activities</b>	<b>98.8</b>	<b>94.2</b>	<b>-4.6</b>	<b>99.6</b>	<b>99.3</b>	<b>-0.3</b>

Table 3: NLU Evaluation Results in F1-scores (%) for DIET+ConveRT Models Trained on Kid Space Home POC Data & Tested on Home Deployment Data

the performance drops we previously observed at school (Okur et al., 2022c). We witness distributional and utterance-length differences between POC/training and deployment/test datasets. Real-world data would always be noisier than anticipated as these utterances come from younger kids playing math games in dynamic conditions.

To further improve the performance of our Kid Space Home NLU models (trained on POC data) by leveraging this recent deployment data, we experiment with merging the two datasets for training and evaluating the performance on individual deployment sessions via leave-one-out (LOO) CV. At each of the 12 runs (for 12 sessions/kids), we merge the POC data with 11 sessions of deployment data for model training and use the remaining session as a test set, then take the average performance of these runs. That would simulate how combining POC with real-world deployment data would help us train more robust NLU models that perform better on unseen data in future deployment sessions. The overall F1-scores reach 96.5% for intents (2.3% gain from 94.2%) and 99.4% for entities (0.1% gain) with LOOCV, which are promising for our future deployments.

To inspect the ASR module of our SLU pipeline, we experiment with Rockhopper, Google, and Whisper-base/small/medium ASR models evaluated on the same audio data collected during home deployments. Using the manual session transcripts as a reference, we compute the average WER for kids with each ASR engine to investigate the most feasible solution. Table 4 summarizes WER results before and after standard pre-processing steps (e.g., lower casing and punctuation removal) as well as application-specific filters (e.g., num2word and cleaning). The numbers are transcribed inconsistently within reference transcripts plus ASR output (e.g., 35 vs. thirty-five), and we need to standardize them all in word forms. The cleaning

ASR Model	Raw Output	Lowercase (LC)	Remove Punct (RP)	Num2Word (NW)	LC & RP	LC & RP & NW	NW & Clean	LC & RP & NW & Clean
Rockhopper	0.939	0.919	0.924	0.937	0.886	0.884	0.937	0.884
Google Cloud	0.829	0.798	0.775	0.763	0.695	0.602	0.763	0.602
Whisper-base	1.042	1.020	0.971	0.985	0.946	0.856	0.622	<b>0.500</b>
Whisper-small	0.834	0.804	0.760	0.756	0.720	0.621	0.537	<b>0.405</b>
Whisper-medium	0.905	0.870	0.824	0.814	0.785	0.675	0.522	<b>0.384</b>

Table 4: ASR Model Results: Avg Word Error Rates (WER) for Child Speech at Kid Space Home Deployment Data

step is applied to Whisper ASR output only due to known issues such as getting stuck in repeat loops and hallucinations (Radford et al., 2022). We seldom observe trash output from Whisper (4-to-7%) having very long transcriptions with non-sense repetitions/symbols, which hugely affect WER due to their length, yet these samples can be easily auto-filtered. Even after these steps, the relatively high error rates can be attributed to many factors related to the characteristics of these recordings (e.g., incidental voice and phrases), very short utterances to be recognized (e.g., binary yes/no answers or stating numbers with one-or-two words), and recognizing kids’ speech in ordinary home environments. Still, the comparative results indicate that Whisper ASR solutions perform better on kids, and we can benefit from increasing the model size from base to small, while small to medium is close.

For SLU pipeline evaluation, we test our highest-performing NLU models on noisy ASR output. Table 5 presents the Intent and Entity Classification results achieved on home deployment data where the DIET+ConveRT models run on varying ASR models output. Note that Voice Activity Detection (VAD) is an integral part of ASR that decides the presence/absence of human speech. We realize that the VAD stage is filtering out a lot of audio chunks with actual kid speech with Rockhopper and Google. Thus, our VAD-ASR nodes can ignore a lot of audio segments with reference transcripts (57.9% for Rokchopper, 49.1% for Google). That is less of an issue with Whisper-base/small/medium, missing 7.1%/5.7%/4.4% of transcribed utterances (often due to filtering very long and repetitive trash Whisper output). When we treat these entirely missed utterances with no ASR output as classification errors for NLU tasks (i.e., missing to predict intent/entities when no speech is detected), we can adjust the F1-scores accordingly to evaluate the VAD-ASR+NLU pipeline. These VAD-adjusted F1-scores are compared in Table 5, aligned with the WER results, where NLU on Whisper ASR

ASR Model	Intent Detection		Entity Extraction	
	F1	Adjusted-F1	F1	Adjusted-F1
Rockhopper	36.7	15.5	82.9	35.0
Google Cloud	<b>78.0</b>	39.7	96.2	49.0
Whisper-base	64.7	60.0	95.4	88.5
Whisper-small	72.2	68.1	96.6	91.1
Whisper-medium	<b>76.5</b>	<b>73.1</b>	<b>98.5</b>	<b>94.1</b>

Table 5: SLU Pipeline Evaluation Results in F1-scores (%) for ASR+NLU and VAD-Adjusted ASR+NLU on Kid Space Home Deployment Data

performs relatively higher than Google and Rockhopper. For enhanced Intent Recognition in real-world deployments with kids, increasing the ASR model size from small to medium could be worth the trouble for Whisper. Yet, the F1 drop is still huge, from 94.2% with NLU to 73.1% with VAD-ASR+NLU, when VAD-ASR errors propagate into the SLU pipeline.

## 5 Error Analysis

For NLU error analysis, Table 6 reveals utterance samples from our Kid Space Home Deployment data with misclassified intents obtained by the DIET+ConveRT models on manual/human transcripts. These language understanding errors illustrate the potential pain points solely related to the NLU model performances, as we are assuming perfect or human-level ASR here by feeding the manually transcribed utterances into the NLU. Such intent prediction errors occur in real-world deployments for many reasons. For example, authentic user utterances can have multiple intents (e.g., “Yeah. Can we have some carrots?” starts with *affirm* and continues with *out-of-scope*). Some utterances can be challenging due to subtle differences between intent classes (e.g., “Ah this is 70, 7.” is submitting a verbal answer with *state-number* but can easily be mixed with *still-counting* too). Moreover, we observe utterances having *colors* and “flowers” within *out-of-scope* (e.g., “Wow, that’s a lot of red flowers.”), which can be confusing for the NLU models trained on cleaner POC datasets.

Sample Kid Utterance	Intent	Prediction
Pepper.	<i>state-name</i>	<i>answer-valid</i>
Wow, that’s a lot of red flowers.	<i>out-of-scope</i>	<i>answer-flowers</i>
None.	<i>state-number</i>	<i>deny</i>
Nothing.	<i>state-number</i>	<i>deny</i>
Yeah. Can we have some carrots?	<i>affirm</i>	<i>out-of-scope</i>
Okay, Do your magic.	<i>affirm</i>	<i>out-of-scope</i>
Maybe tomorrow.	<i>affirm</i>	<i>out-of-scope</i>
He’s a bear.	<i>out-of-scope</i>	<i>answer-valid</i>
I like the idea of a bear	<i>out-of-scope</i>	<i>answer-valid</i>
Oh, 46? Okay.	<i>still-counting</i>	<i>state-number</i>
94. Okay.	<i>still-counting</i>	<i>state-number</i>
Now we have mountains.	<i>out-of-scope</i>	<i>answer-valid</i>
A pond?	<i>out-of-scope</i>	<i>answer-valid</i>
Sorry, I didn’t understand it. Uh, five tens.	<i>state-number</i>	<i>still-counting</i>
Ah this is 70, 7.	<i>state-number</i>	<i>still-counting</i>

Table 6: NLU Error Analysis: Intent Recognition Error Samples from Kid Space Home Deployment Data

Human Transcript	ASR Output	ASR Model	Intent	Prediction
Six.	thanks	Rockhopper	<i>state-number</i>	<i>thank</i>
fifteen	if he	Rockhopper	<i>state-number</i>	<i>out-of-scope</i>
fifteen	Mickey	Google Cloud	<i>state-number</i>	<i>state-name</i>
Five.	bye	Google Cloud	<i>state-number</i>	<i>goodbye</i>
Blue.	Blair.	Whisper-base	<i>state-color</i>	<i>state-name</i>
twenty	Plenty.	Whisper-base	<i>state-number</i>	<i>had-fun-a-lot</i>
A lot.	Oh, la.	Whisper-base	<i>had-fun-a-lot</i>	<i>out-of-scope</i>
A lot.	Oh, wow.	Whisper-small	<i>had-fun-a-lot</i>	<i>out-of-scope</i>
Two.	you	Whisper-small	<i>state-number</i>	<i>out-of-scope</i>
Four.	I’m going to see this floor.	Whisper-small	<i>state-number</i>	<i>out-of-scope</i>
twenty	Swamy?	Whisper-medium	<i>state-number</i>	<i>state-name</i>
Eight.	E.	Whisper-medium	<i>state-number</i>	<i>out-of-scope</i>

Table 7: SLU Pipeline (ASR+NLU): Intent Recognition Error Samples from Kid Space Home Deployment Data

For further error analysis on the SLU pipeline (ASR+NLU), Table 7 demonstrates Intent Recognition error samples from Kid Space Home Deployment data obtained on ASR output with several speech recognition models we explored. These samples depict anticipated error propagation from speech recognition to language understanding modules in the cascaded SLU approach. Please check Appendix A for a more detailed ASR error analysis.

## 6 Conclusion

To increase the quality of math learning experiences at home for early childhood education, we develop a multimodal dialogue system with play-based learning activities, helping the kids gain basic math skills. This study investigates a modular SLU pipeline for kids with cascading ASR and NLU modules, evaluated on our first home deployment data with 12 kids at individual homes. For NLU, we examine the advantages of a multi-task architecture and experiment with numerous pre-

trained language representations for Intent Recognition and Entity Extraction tasks in our application domain. For ASR, we inspect the WER with several solutions that are either low-power and local (e.g., Rockhopper), commercial (e.g., Google Cloud), or open-source (e.g., Whisper) with varying model sizes and conclude that Whisper-medium outperforms the rest on kids’ speech at authentic home environments. Finally, we evaluate the SLU pipeline by running our best-performing NLU models, DIET+ConveRT, on VAD-ASR output to observe the significant effects of cascaded errors due to noisy voice detection and speech recognition performance with kids in realistic home deployment settings. In the future, we aim to fine-tune the Whisper ASR acoustic models on kids’ speech and language models on domain-specific math content. Moreover, we consider exploring N-Best-ASR-Transformers (Ganesan et al., 2021) to leverage multiple Whisper ASR hypotheses and mitigate errors propagated into cascading SLU.

## Limitations

By building this task-specific dialogue system for kids, we aim to increase the overall quality of basic math education and learning at-home experiences for younger children. In our previous school deployments, the overall cost of the whole school/classroom setup, including the wall/ceiling-mounted projector, 3D/RGB-D cameras, LiDAR sensor, wireless lavalier microphones, servers, etc., can be considered as a limitation for public schools and disadvantaged populations. When we shifted our focus to home learning usages after the COVID-19 pandemic, we simplified the overall setup for 1:1 learning with a PC laptop with a built-in camera, a depth camera on a tripod, a lapel mic, and a playmat with cubes and sticks. However, even this minimal instrumentation suitable for home setup can be a limitation for kids with lower socioeconomic status. Moreover, the dataset size of our initial home deployment data collected from 12 kids in 12 sessions is relatively small, with around 12 hours of audio data manually transcribed and annotated. Collecting multimodal data at authentic homes of individual kids within our target age group (e.g., 5-to-8 years old) and labor-intensive labeling process is challenging and costly. To overcome these data scarcity limitations and develop dialogue systems for kids with such small-data regimes, we had to rely on transfer learning approaches as much as possible. However, the dataset sizes affect the generalizability of our explorations, the reliability of some results, and ultimately the robustness of our multimodal dialogue system for deployments with kids in the real world.

## Ethics Statement

Prior to our initial research deployments at home, a meticulous process of Privacy Impact Assessment is pursued. The legal approval processes are completed to operate our research with educators, parents, and the kids. Individual participants and parties involved have signed the relevant consent forms in advance, which inform essential details about our research studies. The intentions and procedures and how the participant data will be collected and utilized to facilitate our research are explained in writing in these required consent forms. Our collaborators comply with stricter data privacy policies as well.

## Acknowledgements

We aspire to share our gratitude and acknowledge our former and current colleagues in the Kid Space team at Intel Labs. Particularly: (i) Hector Coudrier Maruri, Juan Del Hoyo Ontiveros, and Georg Stemmer for developing the VAD-ASR node to obtain the ASR output that we use in our SLU pipeline; (ii) Benjamin Bair, Lenitra Durham, Sai Prasad, Giuseppe Raffa, Celal Savur, and Sangita Sharma for designing the HW/SW architectural setup, developing the Wizard UI and game logic nodes, and supporting data collection; (iii) Ankur Agrawal, Arturo Bringas Garcia, Vishwajeet Narwal, and Guillermo Rivas Aguilar for developing the Student UI via the Unity game engine; (iv) Glen Anderson, Sinem Aslan, Rebecca Chierichetti, Pete Denman, John Sherry, and Meng Shi for conducting UX studies and performing interaction design to conceptualize school and home usages; (v) David Gonzalez Aguirre, Gesem Gudino Mejia, and Julio Zamora Esquivel for developing the visual understanding nodes to support this research. We would also want to gratefully acknowledge our field team members from Summa Linguae Technologies, especially Rick Lin and Brenda Tumbalobos Cubas, for their exceptional support in executing the data collection and transcription/annotation tasks in collaboration with our Intel Labs Kid Space team. Finally, we should thank the Rasa team and community developers for their open-source framework and contributions that empowered us to conduct our research.

## References

- Glen J. Anderson, Selvakumar Panneer, Meng Shi, Carl S. Marshall, Ankur Agrawal, Rebecca Chierichetti, Giuseppe Raffa, John Sherry, Daria Loi, and Lenitra Megail Durham. 2018. *Kid space: Interactive learning in a smart environment*. In *Proceedings of the Group Interaction Frontiers in Technology, GIFT'18*, New York, NY, USA. Association for Computing Machinery.
- Sinem Aslan, Ankur Agrawal, Nese Alyuz, Rebecca Chierichetti, Lenitra M Durham, Ramesh Manuvinakurike, Eda Okur, Saurav Sahay, Sangita Sharma, John Sherry, Giuseppe Raffa, and Lama Nachman. 2022. *Exploring kid space in the wild: a preliminary study of multimodal and immersive collaborative play-based learning experiences*. *Educational Technology Research and Development*, 70:205–230.
- Sinem Aslan, Nese Alyuz, Cagri Tanriover, Sinem E.

- Mete, Eda Okur, Sidney K. D’Mello, and Asli Arslan Esme. 2019. [Investigating the impact of a real-time, multimodal student engagement analytics technology in authentic classrooms](#). In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI ’19, page 1–12, New York, NY, USA. Association for Computing Machinery.
- Zhangir Azerbayev, Bartosz Piotrowski, and Jeremy Avigad. 2022. [Proofnet: A benchmark for autoformalizing and formally proving undergraduate-level mathematics problems](#). In *Workshop MATH-AI: Toward Human-Level Mathematical Reasoning, 36th Conference on Neural Information Processing Systems (NeurIPS 2022)*, New Orleans, Louisiana, USA.
- Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.
- Yu Bai, Ferdy Hubers, Catia Cucchiari, and Helmer Strik. 2021. [An ASR-based Reading Tutor for Practicing Reading Skills in the First Grade: Improving Performance through Threshold Adjustment](#). In *Proc. IberSPEECH 2021*, pages 11–15.
- Yu Bai, Ferdy Hubers, Catia Cucchiari, Roeland van Hout, and Helmer Strik. 2022. [The Effects of Implicit and Explicit Feedback in an ASR-based Reading Tutor for Dutch First-graders](#). In *Proc. Interspeech 2022*, pages 4476–4480.
- Ryan S Baker. 2021. Artificial intelligence in education: Bringing it all together. *OECD Digital Education Outlook 2021: Pushing the Frontiers with Artificial Intelligence, Blockchain and Robots*, pages 43–51.
- Vivek Bhardwaj, Mohamed Tahar Ben Othman, Vinay Kukreja, Youcef Belkhier, Mohit Bajaj, B Srikanth Goud, Ateeq Ur Rehman, Muhammad Shafiq, and Habib Hamam. 2022. Automatic speech recognition (asr) systems for children: A systematic literature review. *Applied Sciences*, 12(9):4419.
- Serge Bibauw, Wim Van den Noortgate, Thomas François, and Piet Desmet. 2022. Dialogue systems for language learning: a meta-analysis. *Language Learning & Technology*, 26(1).
- Nathaniel Blanchard, Michael Brady, Andrew M. Olney, Marci Glaus, Xiaoyi Sun, Martin Nystrand, Borhan Samei, Sean Kelly, and Sidney D’Mello. 2015. A study of automatic speech recognition in noisy classroom environments for automated dialog analysis. In *Artificial Intelligence in Education*, pages 23–33, Cham. Springer International Publishing.
- Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. [Rasa: Open source language understanding and dialogue management](#). In *Conversational AI Workshop, NIPS 2017*.
- Eric Booth, Jake Carns, Casey Kennington, and Nader Rafla. 2020. [Evaluating and improving child-directed automatic speech recognition](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6340–6345, Marseille, France. European Language Resources Association.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, 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.
- Tanja Bunk, Daksh Varshneya, Vladimir Vlasov, and Alan Nichol. 2020. [DIET: lightweight language understanding for dialogue systems](#). *CoRR*, abs/2004.09936.
- Mikhail Burtsev, Alexander Seliverstov, Rafael Airapetyan, Mikhail Arkhipov, Dilyara Baymurzina, Nickolay Bushkov, Olga Gureenkova, Taras Khakhulin, Yuri Kuratov, Denis Kuznetsov, Alexey Litinsky, Varvara Logacheva, Alexey Lymar, Valentin Malykh, Maxim Petrov, Vadim Polulyakh, Leonid Pugachev, Alexey Sorokin, Maria Vikhreva, and Marat Zaynutdinov. 2018. [DeepPavlov: Open-source library for dialogue systems](#). In *Proceedings of ACL 2018, System Demonstrations*, pages 122–127, Melbourne, Australia. Association for Computational Linguistics.
- Aoife Cahill, James H Fife, Brian Riordan, Avijit Vajpayee, and Dmytro Galochkin. 2020. [Context-based automated scoring of complex mathematical responses](#). In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 186–192, Seattle, WA, USA → Online. Association for Computational Linguistics.
- Bernard Cesarone. 2008. Early childhood mathematics: Promoting good beginnings. *Childhood Education*, 84(3):189.
- Ying-Hong Chan, Ho-Lam Chung, and Yao-Chung Fan. 2021. [Improving controllability of educational question generation by keyword provision](#). *CoRR*, abs/2112.01012.
- Maud Chassignol, Aleksandr Khoroshavin, Alexandra Klimova, and Anna Bilyatdinova. 2018. [Artificial intelligence trends in education: a narrative overview](#). *Procedia Computer Science*, 136:16–24. 7th International Young Scientists Conference on Computational Science, YSC2018, 02-06 July 2018, Heraklion, Greece.

- Felix Claus, Hamurabi Gamboa Rosales, Rico Petrick, Horst-Udo Hain, and Rüdiger Hoffmann. 2013. A survey about databases of children’s speech. In *INTERSPEECH*, pages 2410–2414.
- Debby RE Cotton, Peter A Cotton, and J Reuben Shipway. 2023. Chatting and cheating: Ensuring academic integrity in the era of chatgpt. *Innovations in Education and Teaching International*, pages 1–12.
- Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. [Transformer-XL: Attentive language models beyond a fixed-length context](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.
- Debajyoti Datta, Maria Phillips, Jennifer L. Chiu, Ginger S. Watson, James P. Bywater, Laura E. Barnes, and Donald E. Brown. 2020. [Improving classification through weak supervision in context-specific conversational agent development for teacher education](#). *CoRR*, abs/2010.12710.
- 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.
- Richeng Duan and Nancy F Chen. 2020. Unsupervised feature adaptation using adversarial multi-task training for automatic evaluation of children’s speech. In *INTERSPEECH*, pages 3037–3041.
- Satwik Dutta, Dwight Irvin, Jay Buzhardt, and John H.L. Hansen. 2022. [Activity focused speech recognition of preschool children in early childhood classrooms](#). In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 92–100, Seattle, Washington. Association for Computational Linguistics.
- Jean-Claude Falmagne, Dietrich Albert, Christopher Doble, David Eppstein, and Xiangen Hu. 2013. *Knowledge spaces: Applications in education*. Springer Science & Business Media.
- Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. [Language-agnostic BERT sentence embedding](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.
- Walter Chang Franck Dernoncourt, Trung Bui. 2018. A framework for speech recognition benchmarking. In *Interspeech*.
- Simon Frieder, Luca Pinchetti, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Christian Petersen, Alexis Chevalier, and Julius Berner. 2023. Mathematical capabilities of chatgpt. *arXiv preprint arXiv:2301.13867*.
- Karthik Ganesan, Pakhi Bamdev, Jaivarsan B, Amresh Venugopal, and Abhinav Tushar. 2021. [N-best ASR transformer: Enhancing SLU performance using multiple ASR hypotheses](#). 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 93–98, Online. Association for Computational Linguistics.
- Matteo Gerosa, Diego Giuliani, and Fabio Brugnara. 2007. Acoustic variability and automatic recognition of children’s speech. *Speech Communication*, 49(10-11):847–860.
- Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. [Slot-gated modeling for joint slot filling and intent prediction](#). 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 753–757, New Orleans, Louisiana. Association for Computational Linguistics.
- A.C. Graesser, P. Chipman, B.C. Haynes, and A. Olney. 2005. [Autotutor: an intelligent tutoring system with mixed-initiative dialogue](#). *IEEE Transactions on Education*, 48(4):612–618.
- Joshua Grossman, Zhiyuan Lin, Hao Sheng, Johnny Tian-Zheng Wei, Joseph J Williams, and Sharad Goel. 2019. Mathbot: Transforming online resources for learning math into conversational interactions. *AAAI 2019 Story-Enabled Intelligence*.
- Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulić. 2020. [ConveRT: Efficient and accurate conversational representations from transformers](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2161–2174, Online. Association for Computational Linguistics.
- Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. [spaCy: Industrial-strength natural language processing in python](#).
- Shifeng Huang, Jiawei Wang, Jiao Xu, Da Cao, and Ming Yang. 2021. [Real2: An end-to-end memory-augmented solver for math word problems](#). In *Workshop on Math AI for Education (MATHAI4ED), 35th Conference on Neural Information Processing Systems (NeurIPS 2021)*.
- Jiyou Jia, Yunfan He, and Huixiao Le. 2020. A multi-modal human-computer interaction system and its application in smart learning environments. In *Blended*

- Learning. Education in a Smart Learning Environment*, pages 3–14, Cham. Springer International Publishing.
- Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günemann, Eyke Hüllermeier, et al. 2023. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103:102274.
- Amelia C Kelly, Eleni Karamichali, Armin Saeb, Karel Veselý, Nicholas Parslow, Agape Deng, Arnaud Letondor, Robert O’Regan, and Qiru Zhou. 2020. Soapbox labs verification platform for child speech. In *INTERSPEECH*, pages 486–487.
- John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, ICML, pages 282–289.
- Siddique Latif, Aun Zaidi, Heriberto Cuayahuitl, Fahad Shamshad, Moazzam Shoukat, and Junaid Qadir. 2023. Transformers in speech processing: A survey. *arXiv preprint arXiv:2303.11607*.
- Sweta P Lende and MM Raghuvanshi. 2016. Question answering system on education acts using nlp techniques. In *2016 world conference on futuristic trends in research and innovation for social welfare (Startup Conclave)*, pages 1–6. IEEE.
- Bing Liu and Ian Lane. 2016. [Attention-based recurrent neural network models for joint intent detection and slot filling](#). In *Interspeech 2016*, pages 685–689.
- Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021a. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.
- Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2021b. Benchmarking natural language understanding services for building conversational agents. In *Increasing Naturalness and Flexibility in Spoken Dialogue Interaction: 10th International Workshop on Spoken Dialogue Systems*, pages 165–183. Springer.
- 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](#). *CoRR*, abs/1907.11692.
- Ekaterina Logina and Dries Benoit. 2022. Structural information in mathematical formulas for exercise difficulty prediction: a comparison of nlp representations. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 101–106.
- Jakub Macina, Nico Daheim, Lingzhi Wang, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023. Opportunities and challenges in neural dialog tutoring. *arXiv preprint arXiv:2301.09919*.
- Andrea Madotto, Zihan Liu, Zhaojiang Lin, and Pascale Fung. 2020. Language models as few-shot learner for task-oriented dialogue systems. *arXiv preprint arXiv:2008.06239*.
- Behrooz Mansouri, Vít Novotný, Anurag Agarwal, Douglas W. Oard, and Richard Zanibbi. 2022. Overview of arqmath-3 (2022): Third clef lab on answer retrieval for questions on math. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 286–310, Cham. Springer International Publishing.
- Behrooz Mansouri, Shaurya Rohatgi, Douglas W Oard, Jian Wu, C Lee Giles, and Richard Zanibbi. 2019. Tangent-cft: An embedding model for mathematical formulas. In *Proceedings of the 2019 ACM SIGIR international conference on theory of information retrieval*, pages 11–18.
- Shikib Mehri, Mihail Eric, and Dilek Hakkani-Tür. 2020. [Dialoglue: A natural language understanding benchmark for task-oriented dialogue](#). *CoRR*, abs/2009.13570.
- Nrupatunga, Aashish Kumar, and Anoop Rajagopal. 2021. [Phygital math learning with handwriting for kids](#). In *Workshop on Math AI for Education (MATHAI4ED), 35th Conference on Neural Information Processing Systems (NeurIPS 2021)*.
- Benjamin D Nye, Philip I Pavlik, Alistair Windsor, Andrew M Olney, Mustafa Hajeer, and Xiangen Hu. 2018. Skope-it (shareable knowledge objects as portable intelligent tutors): overlaying natural language tutoring on an adaptive learning system for mathematics. *International journal of STEM education*, 5:1–20.
- Chinedu Wilfred Okonkwo and Abejide Ade-Ibijola. 2021. Chatbots applications in education: A systematic review. *Computers and Education: Artificial Intelligence*, 2:100033.
- Eda Okur, Shachi H. Kumar, Saurav Sahay, Asli Arslan Esme, and Lama Nachman. 2019. [Natural language interactions in autonomous vehicles: Intent detection and slot filling from passenger utterances](#). In *Computational Linguistics and Intelligent Text Processing*, pages 334–350, Cham. Springer Nature Switzerland.
- Eda Okur, Saurav Sahay, Roddy Fuentes Alba, and Lama Nachman. 2022a. [End-to-end evaluation of a spoken dialogue system for learning basic mathematics](#). In *Proceedings of the 1st Workshop on Mathematical Natural Language Processing (MathNLP)*, pages 51–64, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.



- Eda Okur, Saurav Sahay, and Lama Nachman. 2022b. [Data augmentation with paraphrase generation and entity extraction for multimodal dialogue system](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4114–4125, Marseille, France. European Language Resources Association.
- Eda Okur, Saurav Sahay, and Lama Nachman. 2022c. [NLU for game-based learning in real: Initial evaluations](#). In *Proceedings of the 9th Workshop on Games and Natural Language Processing within the 13th Language Resources and Evaluation Conference*, pages 28–39, Marseille, France. European Language Resources Association.
- OpenAI. 2022. Chatgpt: Optimizing language models for dialogue.
- Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.
- Shuai Peng, Ke Yuan, Liangcai Gao, and Zhi Tang. 2021. [Mathbert: A pre-trained model for mathematical formula understanding](#). *CoRR*, abs/2105.00377.
- Ana Cristina Pires, Fernando González Perilli, Ewelina Bakala, Bruno Fleisher, Gustavo Sansone, and Sebastián Marichal. 2019. [Building blocks of mathematical learning: Virtual and tangible manipulatives lead to different strategies in number composition](#). *Frontiers in Education*, 4.
- Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. 2011. The kaldı speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, CONF. IEEE Signal Processing Society.
- Srikrishna Raamadhurai, Ryan Baker, and Vikraman Poduval. 2019. [Curio SmartChat : A system for natural language question answering for self-paced k-12 learning](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 336–342, Florence, Italy. Association for Computational Linguistics.
- Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*.
- 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.
- Manav Rathod, Tony Tu, and Katherine Stasaski. 2022. [Educational multi-question generation for reading comprehension](#). In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 216–223, Seattle, Washington. Association for Computational Linguistics.
- Kenneth Reeder, Jon Shapiro, Jane Wakefield, and Reg D’Silva. 2015. Speech recognition software contributes to reading development for young learners of english. *International Journal of Computer-Assisted Language Learning and Teaching (IJCALLT)*, 5(3):60–74.
- Anja Reusch, Maik Thiele, and Wolfgang Lehner. 2022. Transformer-encoder and decoder models for questions on math. *Proceedings of the Working Notes of CLEF 2022*, pages 5–8.
- Roberto Reyes, David Garza, Leonardo Garrido, Víctor De la Cueva, and Jorge Ramirez. 2019. Methodology for the implementation of virtual assistants for education using google dialogflow. In *Advances in Soft Computing: 18th Mexican International Conference on Artificial Intelligence, MICAI 2019, Xalapa, Mexico, October 27–November 2, 2019, Proceedings 18*, pages 440–451. Springer.
- J. Elizabeth Richey, Jiayi Zhang, Rohini Das, Juan Miguel Andres-Bray, Richard Scruggs, Michael Mogessie, Ryan S. Baker, and Bruce M. McLaren. 2021. Gaming and confrustion explain learning advantages for a math digital learning game. In *Artificial Intelligence in Education*, pages 342–355, Cham. Springer International Publishing.
- Lars Rumberg, Hanna Ehlert, Ulrike Lüdtke, and Jörn Ostermann. 2021. Age-invariant training for end-to-end child speech recognition using adversarial multi-task learning. In *Interspeech*, pages 3850–3854.
- Saurav Sahay, Shachi H. Kumar, Eda Okur, Haroon Syed, and Lama Nachman. 2019. [Modeling intent, dialog policies and response adaptation for goal-oriented interactions](#). In *Proceedings of the 23rd Workshop on the Semantics and Pragmatics of Dialogue - Full Papers*, London, United Kingdom. SEM-DIAL.
- Saurav Sahay, Eda Okur, Nagib Hakim, and Lama Nachman. 2021. [Semi-supervised interactive intent labeling](#). In *Proceedings of the Second Workshop on Data Science with Human in the Loop: Language Advances*, pages 31–40, Online. Association for Computational Linguistics.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. [Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter](#). In *5th EMC2 Workshop - Energy Efficient Training and Inference of Transformer Based Models, 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*.
- Iulian Vlad Serban, Ryan Lowe, Peter Henderson, Laurent Charlin, and Joelle Pineau. 2018. A survey of

- available corpora for building data-driven dialogue systems: The journal version. *Dialogue & Discourse*, 9(1):1–49.
- Jia Tracy Shen, Michiharu Yamashita, Ethan Prihar, Neil T. Heffernan, Xintao Wu, and Dongwon Lee. 2021. [Mathbert: A pre-trained language model for general NLP tasks in mathematics education](#). *CoRR*, abs/2106.07340.
- Prashanth Gurunath Shivakumar and Panayiotis Georgiou. 2020. Transfer learning from adult to children for speech recognition: Evaluation, analysis and recommendations. *Computer speech & language*, 63:101077.
- Prashanth Gurunath Shivakumar, Alexandros Potamianos, Sungbok Lee, and Shrikanth Narayanan. 2014. [Improving speech recognition for children using acoustic adaptation and pronunciation modeling](#). In *Fourth Workshop on Child Computer Interaction (WOCCI 2014)*.
- Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, et al. 2022. Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage. *arXiv preprint arXiv:2208.03188*.
- Kayleigh Skene, Christine M O’Farrelly, Elizabeth M Byrne, Natalie Kirby, Eloise C Stevens, and Paul G Ramchandani. 2022. Can guidance during play enhance children’s learning and development in educational contexts? a systematic review and meta-analysis. *Child Development*.
- Georg Stemmer, Munir Georges, Joachim Hofer, Piotr Rozen, Josef G Bauer, Jakub Nowicki, Tobias Bocklet, Hannah R Colett, Ohad Falik, Michael Deisher, et al. 2017. Speech recognition and understanding on hardware-accelerated dsp. In *Interspeech*, pages 2036–2037.
- Georg Stemmer, Christian Hacker, Stefan Steidl, and Elmar Nöth. 2003. Acoustic normalization of children’s speech. In *Eighth European Conference on Speech Communication and Technology*.
- Yueqiu Sun, Tangible Play, Rohitkrishna Nambiar, and Vivek Vidyasagan. 2021. [Gamifying math education using object detection](#). In *Workshop on Math AI for Education (MATHAI4ED)*, 35th Conference on Neural Information Processing Systems (NeurIPS 2021).
- Abhijit Suresh, Jennifer Jacobs, Charis Harty, Margaret Perkoff, James H. Martin, and Tamara Sumner. 2022a. [The TalkMoves dataset: K-12 mathematics lesson transcripts annotated for teacher and student discursive moves](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4654–4662, Marseille, France. European Language Resources Association.
- Abhijit Suresh, Jennifer Jacobs, Margaret Perkoff, James H. Martin, and Tamara Sumner. 2022b. [Fine-tuning transformers with additional context to classify discursive moves in mathematics classrooms](#). In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 71–81, Seattle, Washington. Association for Computational Linguistics.
- Anais Tack and Chris Piech. 2022. The ai teacher test: Measuring the pedagogical ability of blender and gpt-3 in educational dialogues. In *Proceedings of the 15th International Conference on Educational Data Mining*, page 522.
- Kaveh Taghipour and Hwee Tou Ng. 2016. [A neural approach to automated essay scoring](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1891, Austin, Texas. Association for Computational Linguistics.
- Elka Torpey. 2012. Math at work: Using numbers on the job. *Occupational Outlook Quarterly*, 56(3):2–13.
- Gladys Tyen, Mark Brenchley, Andrew Caines, and Paula Buttery. 2022. [Towards an open-domain chatbot for language practice](#). In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 234–249, Seattle, Washington. Association for Computational Linguistics.
- Jonathan Uesato, Nate Kushman, Ramana Kumar, H Francis Song, Noah Yamamoto Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. [Solving math word problems with process-based and outcome-based feedback](#). In *Workshop MATH-AI: Toward Human-Level Mathematical Reasoning, 36th Conference on Neural Information Processing Systems (NeurIPS 2022)*, New Orleans, Louisiana, USA.
- Andrea Vanzo, Emanuele Bastianelli, and Oliver Lemon. 2019. [Hierarchical multi-task natural language understanding for cross-domain conversational AI: HERMIT NLU](#). In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 254–263, Stockholm, Sweden. Association 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 *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.
- Thiemo Wambsganss, Rainer Winkler, Matthias Söllner, and Jan Marco Leimeister. 2020. A conversational agent to improve response quality in course evaluations. In *Extended Abstracts of the 2020 CHI conference on human factors in computing systems*, pages 1–9.

- Liyun Wen, Xiaojie Wang, Zhenjiang Dong, and Hong Chen. 2018. Jointly modeling intent identification and slot filling with contextual and hierarchical information. In *Natural Language Processing and Chinese Computing*, pages 3–15, Cham. Springer International Publishing.
- Rainer Winkler, Sebastian Hobert, Antti Salovaara, Matthias Söllner, and Jan Marco Leimeister. 2020. Sara, the lecturer: Improving learning in online education with a scaffolding-based conversational agent. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pages 1–14.
- Rainer Winkler and Matthias Söllner. 2018. [Unleashing the potential of chatbots in education: A state-of-the-art analysis](#). In *Academy of Management Annual Meeting (AOM)*.
- Sebastian Wollny, Jan Schneider, Daniele Di Mitri, Joshua Weidlich, Marc Rittberger, and Hendrik Drachler. 2021. Are we there yet?-a systematic literature review on chatbots in education. *Frontiers in artificial intelligence*, 4:654924.
- Fei Wu, Leibny Paola García-Perera, Daniel Povey, and Sanjeev Khudanpur. 2019. Advances in automatic speech recognition for child speech using factored time delay neural network. In *Interspeech*, pages 1–5.
- Ledell Wu, Adam Fisch, Sumit Chopra, Keith Adams, and Antoine Bordes Jason Weston. 2018. Starspace: Embed all the things! In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI’18/IAAI’18/EAAI’18. AAAI Press.
- Zhicheng Yang, Jinghui Qin, Jiaqi Chen, Liang Lin, and Xiaodan Liang. 2022. [LogicSolver: Towards interpretable math word problem solving with logical prompt-enhanced learning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1–13, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. [Xlnet: Generalized autoregressive pretraining for language understanding](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Gary Yeung and Abeer Alwan. 2018. On the difficulties of automatic speech recognition for kindergarten-aged children. *Interspeech 2018*.
- Gary Yeung, Ruchao Fan, and Abeer Alwan. 2021. Fundamental frequency feature normalization and data augmentation for child speech recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6993–6997. IEEE.
- Xuesong Zhai, Xiaoyan Chu, Ching Sing Chai, Morris Siu Yung Jong, Andreja Istenic, Michael Spector, Jia-Bao Liu, Jing Yuan, and Yan Li. 2021. A review of artificial intelligence (AI) in education from 2010 to 2020. *Complexity*, 2021.
- Xiaodong Zhang and Houfeng Wang. 2016. [A joint model of intent determination and slot filling for spoken language understanding](#). In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI’16*, pages 2993–2999. AAAI Press.
- Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. [DIALOGPT : Large-scale generative pre-training for conversational response generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.
- Zheng Zhang, Ying Xu, Yanhao Wang, Bingsheng Yao, Daniel Ritchie, Tongshuang Wu, Mo Yu, Dakuo Wang, and Toby Jia-Jun Li. 2022. Storybuddy: A human-ai collaborative chatbot for parent-child interactive storytelling with flexible parental involvement. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–21.

## A Appendix: Additional Error Analysis

Please refer to Table 8 for additional error analysis on ASR output from our home deployment data. Here, we compare manually transcribed utterances (i.e., human transcripts) with the speech recognition output (i.e., raw ASR transcripts) using five different ASR models that we investigated in this study. These ASR errors demonstrate the challenges faced in the speech recognition model performances on kids’ speech, which potentially would be propagated into the remaining modules in the conventional task-oriented dialogue pipeline.

We may attribute various factors to these speech recognition errors, often related to our deployment data characteristics. Incidental voices and phrases constitute a good chunk of the overall home deployment data, along with very short utterances to be recognized (e.g., stating names, colors, types of flowers, numbers, and binary answers with one-or-two words), plus the remaining known challenges present with recognizing kids’ speech in noisy real-world environments.

Human Transcript	Rockhopper	Google Cloud	Whisper-base	Whisper-small	Whisper-medium
Atticus.	-	-	Yeah, that's cute.	I have a kiss.	Now I have to kiss.
I am Genevieve.	i'm twenty-two	I'm going to be	I'm Kennedy.	I'm Genevieve.	I'm Genevieve.
Red.	rab	-	Ralph.	Red.	Red.
Blue.	lil	blue	Blair.	Blue.	Blue.
Yes,	laughs	yes	Yes?	Yes?	Yes?
Roses.	it is	roses	Okay.	Okay	focus
Zero.	you know	no	No.	No, no.	No.
four.	you swore	-	forward.	Over.	Over.
five.	-	bye	Bye.	Bye.	Bye.
eight	all	-	Thank you.	Bye.	Oh
forty eight	wall e	48	48	48	48
forty nine	already	49	49	49	49
fifty one	if you want	51	51	51	51
seventy four	stopping before	74	74	74	74
Maybe tomorrow.	novarro	tomorrow	I need some water, though.	I'm going to leave it tomorrow.	I'm leaving tomorrow.
Flowers, flowers in the greenhouse?	lean forward phelps hours than we	Greenhouse	In forward, in forward, in the green house.	I think forward, both flowers and the greenhouse.	In the green house.
There are seventeen, and seventeen minus ten equals seven.	seventeen seventeen rooms	17 + 17 - 27	There are 17 and 17 minus 10 equals 7.	There are 17 and 17 minus 10 equals 7.	What is the maximum number of children in the world? Um... There are 17 and 17 minus 10 equals 7.

Table 8: ASR Error Samples from Kid Space Home Deployment Data

# Socratic Questioning of Novice Debuggers: A Benchmark Dataset and Preliminary Evaluations

Erfan Al-Hossami<sup>1</sup>, Razvan Bunescu<sup>1</sup>, Ryan Teehan<sup>2</sup>, Laurel Powell<sup>1</sup>,  
Khyati Mahajan<sup>1</sup>, and Mohsen Dorodchi<sup>1</sup>

<sup>1</sup>University of North Carolina at Charlotte, Charlotte, NC

<sup>2</sup>New York University, New York City, NY

{ealhossa, rbunescu}@uncc.edu

## Abstract

Socratic questioning is a teaching strategy where the student is guided towards solving a problem on their own, instead of being given the solution directly. In this paper, we introduce a dataset of Socratic conversations where an instructor helps a novice programmer fix buggy solutions to simple computational problems. The dataset is then used for benchmarking the Socratic debugging abilities of GPT-based language models. While GPT-4 is observed to perform much better than GPT-3.5, its precision, and recall still fall short of human expert abilities, motivating further work in this area.

🔗 <https://github.com/taisazero/socratic-debugging-benchmark>

## 1 Introduction and Motivation

Educational needs for computer science (CS) are on the rise, due to increased enrollments in CS programs (Camp et al., 2017). Higher education institutions in particular are affected by the lack of sufficient instructional staff, often resorting to hiring undergraduate Teaching Assistants (TAs) in their computer science courses. An effective TA benefits students by providing timely feedback and assistance that is tailored to each student’s level of proficiency, with measurable and significant impact on student retention rates (Mirza et al., 2019). In practice, however, not all educational institutions benefit uniformly from their TAs. Depending on class sizes and TA allocations, it is often the case that a teaching assistant cannot spend their time equally with all students who need help, especially when nearing office hours or an assignment deadline. Moreover, students who lack fundamental knowledge from prerequisite courses consume significant TA time throughout the course. This comes at a time when there is also a shortage of K-12 computer science teachers, a lack of appropriate training for K-12 educators interested in teaching

CS effectively (Yadav et al., 2016), and rising TA and peer instruction demand in flipped computer science classrooms (Maher et al., 2015).

Overall, the lack of instructional staff, ranging from TAs to K-12 teachers and college educators, motivates the automation of various types of teaching tasks by leveraging the increasing capabilities of AI models, especially in terms of understanding and generating language and code. Prior work in AI for programming education is primarily composed of intelligent tutoring systems (ITS) and learning support systems for programming courses. While some ITS systems allow interactions with a learner through a chat interface (Hobert, 2019), the range of interactions is often limited, as tutoring systems typically focus on giving hints constructed for predefined solutions or predefined Socratic utterances that are specific to a known set of programming exercises (Jeuring et al., 2014; Gerdes et al., 2017; Hobert, 2019; Alshaikh et al., 2020b). Consequently, traditional ITS systems in the programming domain do not generalize to new courses or new coding assignments without human intervention. This situation is however rapidly changing, due to the substantial leaps in performance exhibited by large language models recently, on a wide array of problems. Language models are now capable of solving introductory programming exercises (Hendrycks et al., 2021; Chen et al., 2021) including custom problems created by instructors (Finnie-Ansley et al., 2022). Furthermore, solutions generated by these models are unique and can fool plagiarism software such as MOSS (Biderman and Raff, 2022), presenting educators with further challenges in maintaining academic integrity.

In Socratic questioning, a teacher assists a learner trying to solve a problem beyond their zone of proximal development (Quintana et al., 2004). Language Models (LMs) have been used effectively for generating a particular type of Socratic questions for solving word math problems, wherein

they leverage the sequential structure of steps that compose the solution (Shridhar et al., 2022). Other applications of LMs include automated feedback on student code submissions (Wu et al., 2021), as well as generating programming exercises, unit tests, and code explanations (Sarsa et al., 2022). However, there still remains a substantial gap in leveraging LMs effectively for guiding novice programmers through a coding exercise in a way that maximizes their learning outcomes, similar to how an effective, experienced TA would guide a beginner programmer. For Socratic questioning, in particular, the difficulty of building an effective system is compounded by the scarcity of examples, whereas the limited data that can be found (Chen et al., 2011) does not have sufficient structure to enable the automatic evaluation of Socratic questioning systems.

In this paper, we focus on the task of Socratic questioning for debugging (Wilson, 1987), or Socratic debugging, defined as a conversation between a knowledgeable programmer and a beginner student who comes for help fixing a buggy solution for a simple computational problem (Section 2). To enable the development and evaluation of LM-based instructional agents, we introduce a manually created dataset of dialogues where the main objective is for the student to repair their buggy code themselves by leveraging guidance received from the instructor at every turn (Section 3). However, as originally observed by Wilson (1987), "no precise formula, or line of questioning" is needed to achieve the goals of Socratic questioning. Furthermore, depending also on their expectations with respect to the student's abilities, an instructor can often think of multiple ways of guiding the student at any particular turn in the conversation, leading to a very large space of possible dialogues. Socratic questions lie in a continuum ranging from providing direct hints that give out the answer to offering minimal guidance, enabling instructors to pose queries at an appropriate level that challenges the student while remaining within each student's ability to answer. To facilitate the automatic evaluation and benchmarking of future Socratic questioning systems in terms of their precision and recall, the dataset contributors are asked to provide all alternative utterances that they think could help the student, at every turn in the conversation. This is a currently ongoing, cognitively demanding data generation effort, requiring contributors with sub-

stantial experience in tutoring beginner programmers. We use the current version of the dataset, containing 86 main conversations, to benchmark the Socratic debugging abilities of two large language models in the GPT family, namely GPT-3.5 and GPT-4 (Section 4), noticing a large discrepancy in performance in favor of the more recent GPT-4. We conclude the paper with related work and limitations.

## 2 Task Definition

We formulate the Socratic debugging task as a dyadic conversation between a Student and an Instructor. In this scenario, the Student is assumed to be a beginner programmer who has recently started learning how to code in Python. As part of his<sup>1</sup> learning to code curriculum, the Student is given a coding problem for which he needs to write a function implementing the specified input-to-output relationship. The Student writes the code for the function, however, the code is buggy and he cannot make progress on his own without help, therefore he seeks help from the Instructor. The Instructor is assumed to be a proficient programmer in Python with experience in teaching novice programmers how to code. When contacted by a Student for help, her main aim is to maximize the learning outcomes by following a Socratic guidance approach through which, over one or more dialogue turns, she helps the students figure out where the bug is and how to fix it on their own.

### 2.1 Input

Since the focus of this work is on generating Socratic guidance and not bug identification or fixing bugs, we assume that the AI agent implementing the Instructor also has access to a description of the bug and of one or more bug fixes. The decision to separate Socratic advice generation from bug identification and debugging was motivated by the fact that these subordinate tasks can already be solved efficiently by large LMs with high accuracy. Therefore, at the start of each conversation, we assume the Instructor has access to the *problem description*, a number of *test cases*, the student's *buggy code*, the *bug description*, and one or more *bug fixes*, as shown below in a sample from our dataset. At each turn in the conversation, the Instructor's task is to generate Socratic guidance in response to the Student's current progress in addressing the

<sup>1</sup>The genders were selected at random by tossing a coin.

bug. Consequently, we assume that the Instructor is also given as input a history of the conversation so far, ending with the last utterance from the student. Shown below is an example ending with the second turn from the student, where the turn number is indicated between brackets.

➤ **Problem description:**

Write a function `factorial(n)` that computes the factorial  $n!$  of a natural number  $n$ , which is defined mathematically as:

$$0! = 1$$
$$n! = n \times (n - 1)!$$

Additionally, if the input integer  $n$  is negative the function should return 0.

➤ **Test cases:**

```
assert factorial(-1) == 0
assert factorial(0) == 1
assert factorial(1) == 1
assert factorial(2) == 2
assert factorial(3) == 6
assert factorial(4) == 24
assert factorial(5) == 120
```

➤ **Buggy code:**

```
1. def factorial(n):
2.     if n < 0:
3.         return 0
4.     fact = 1
5.     for i in range(n):
6.         fact = fact * i
7.     return fact
```

➤ **Bug description:**

On line 6, `fact` is multiplied with 0 in the first iteration of the for loop. Consequently, at every iteration `fact` stays equal with 0 instead of being updated to be equal with factorial of  $(i + 1)$ . Therefore, the function will return 0, irrespective of  $n$ .

➤ **Bug fixes:**

1. Replace `i` with `(i + 1)` on line 6.
2. Replace `range(n)` with `range(1, n + 1)` on line 5.

To summarize, the input for the Instructor agent consists of:

1. The **problem description**, a number of **test cases**, the student's **buggy code**, the **bug description**, and one or more **bug fixes**.
2. The **conversation so far**, ending with the last turn from the Student.

➤ **Conversation so far:**

[1] STUDENT: Hi! I implemented the factorial function but it doesn't work and I do not know why. Can you help?

[1] INSTRUCTOR: Sure. Can you tell me for what values of  $n$  it fails and what values it returns in those cases?

[2] STUDENT: For  $n = 1$  or larger it returns the same value, 0.

[2] INSTRUCTOR: *<Socratic guidance>*

## 2.2 Output

Using the input data described above, the Instructor is expected to generate Socratic guidance appropriate for the current state of the conversation, as shown below.

➤ **Socratic guidance:**

Main responses:

Let's see what happens when  $n$  is 1. What is the first value that is assigned to variable `i` in line 5?

Alternative responses:

1. Let's see what happens when  $n$  is 1. Before line 6 is evaluated in the first iteration of the for loop, what are the values of the variables `fact` and `i`?
2. Let's see what happens when  $n$  is 1. Can you insert a new line between lines 5 and 6 that prints the values of the variables `fact` and `i`?
3. Let's see what happens when  $n$  is 1. What does `range(n)` do when  $n$  is 1?
4. Can you tell me what `range(n)` does?

The example above shows a total of 5 Socratic responses, partitioned into 1 main response and 4 alternative responses. Most of the time there are different ways of guiding the student, and ideally, the Instructor should be able to generate all different types of Socratic guidance that are different from each other in non-trivial ways. For example, the

4th alternative focuses the student on correcting the potential misuse of the `range` function, whereas the main response provides a different kind of guidance wherein the student is expected to first notice the wrong code behavior that is caused by the misuse of `range`. Further justification for the decision to include alternative responses will be provided in Section 3 when introducing the data contribution guidelines. Note that only the main response is used to create the history of the conversation so far that is used as input for generating future Instructor turns.

### 3 Benchmark Dataset

To facilitate the development of conversational agents that act under the task definition above, we manually created a dataset of dialogues where a student fixes buggy code on his own by leveraging the Socratic guidance received from an instructor. The dataset is created by sequentially specifying the *Coding problem*  $\rightarrow$  *Bugs*  $\rightarrow$  *Conversations*  $\rightarrow$  *Threads*. First, a coding problem is selected, normally a simple coding exercise situated at a novice level of coding proficiency, such as `Factorial` or `Fibonacci`. The coding problem is specified through the problem description and the associated test cases. Next, one or more buggy implementations are created, with the constraint that each implementation contains exactly one bug. The bugs were selected to reflect common types of mistakes that beginner programmers make, such as forgetting that indexing of sequences starts at 0, boundary bugs, operator misuse, or misunderstanding of basic programming constructs.

For each buggy implementation, a main conversation is created, where a fictional Student, the author of the buggy code, interacts with a fictional Instructor. The aim of the instructor is to guide the student to discover the cause of the bug and fix it on his own through Socratic dialogue. The dialogue always starts with a student utterance. The instructor and the student then take turns in a dialogue, until the bug is successfully fixed. At each turn, the student may also provide a block of code if he made edits to the code at that turn.

Following research in dialogue systems (Gupta et al., 2019), we create multiple reference instructor utterances at each turn. The Main utterance may be optionally followed by one or more Alternative utterances. Given that the aim of this dataset is to benchmark the ability of an artificial

Problems	23
Bugs	34
Dialogues	86
Student turns	537
Student utterances	763
Instructor Turns	497
Instructor utterances	1,329
Total turns	1,034
Total utterances	2,092

Table 1: Summary of the benchmark dataset: Number of programming problems, bugs, dialogues (including all threads), turns, and total utterances (main and alternatives) for both roles (student and instructor).

Instructor agent to generate Socratic guidance, it is especially important that the contributed main and alternative utterances for the Instructor comprehensively explore the entire range of Socratic advice at that point in the conversation. These alternative utterances should be semantically distinct in a non-trivial manner; in particular, they should not be mere paraphrases of the main utterance or of each other. Upon inspection of the conversations created manually, we discovered that one contributor used a vending machine as an analogy to guide the user to conclude that `print` is not the same as `return`. While using analogies can substantially enhance the impact of Socratic questioning, it can lead to an open-ended range of alternatives, as the number of possible analogies is virtually infinite. Since our aim is to create a dataset that can be used to estimate both the recall and precision of a Socratic guidance generator, at this stage we decided to require that Socratic utterances be *literal*, leaving the generation of figurative utterances as a direction for future work. For the Student, alternative utterances may give different or conflicting answers to an Instructor question, reflecting different levels of understanding. Students may give correct or incorrect answers; they may also introduce new bugs when fixing the original bug.

Once the main conversation ends with the student successfully correcting their code and passing all test cases, the contributors are instructed to create up to three conversational threads.

The dialogues in the dataset were created by 10 contributors with extensive experience in CS education as instructors, teaching assistants, or tutors. The starting problems and buggy implementations were selected to contain a variety of syntactic



Language Model	Manual			BLEU-4			BERT F1			Rouge-L		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
GPT-3.5	19.8	31.5	24.3	3.2	1.9	1.8	56.0	38.6	37.4	21.0	13.3	12.8
GPT-4	52.9	50.0	51.4	3.2	5.6	3.8	35.4	63.3	42.0	14.1	24.9	16.7

Table 2: Preliminary evaluation of GPT-3.5 (`gpt-3.5-turbo`) and GPT-4 on our benchmark dataset. Manual evaluation is performed on all instructor turns from a sample of 5 dialogues, whereas automatic evaluation is performed on the entire dataset. We report the Precision (P), Recall (R), and F1 for the manual evaluation, and BLEU-4, BERT F1, and Rouge-L for the automatic evaluation. All results are percentages (%).

and semantic mistakes that are frequently made by novice Python programmers.

To streamline and standardize the collection of Socratic dialogues and code edits for each input problem description and buggy implementation, we developed a 7-page web application using the Streamlit<sup>2</sup> and gsheetsdb<sup>3</sup> libraries. The application guides contributors through selecting a bug, creating initial and conversational threads, and reviewing and submitting their work. During the process, contributors can add main and alternative utterances, undo actions, and edit the chat history. The application also allows importing and exporting dialogues in a standardized form for review. For more details and images of the application, the reader is referred to Appendix A.

## 4 Experimental Evaluations

We evaluate the GPT-3.5 (OpenAI, 2022) and GPT-4 (OpenAI, 2023) language models in terms of their capacity to generate, at each instructor turn, Socratic utterances that match those contributed in the benchmark dataset. Each test example is composed of an input prompt to the language model containing: a steering prompt for Socratic questioning adapted from the GPT-4 blog post<sup>4</sup>, the problem description, the buggy code, the bug description, the bug fixes, the unit tests, the dialogue history so far, and an instruction to the language model to generate all possible semantically distinct Socratic utterances, as shown below.

Respond to the user with all possible distinct Socratic utterances that guide the user to discover and fix the bug described between `<bug_desc>` and `</bug_desc>`. Student code is written between `<code>` and `</code>`

<sup>2</sup><https://streamlit.io/>

<sup>3</sup><https://github.com/betodealmeida/gsheets-db-api>

<sup>4</sup><https://openai.com/research/gpt-4>

throughout the conversation. Utterances that have the same meaning but different words are considered duplicates. Assume that the student has run the test cases.

The list of utterances generated by the LM is then used to estimate precision and recall. After conducting a preliminary, qualitative evaluation of various prompts and instructions we select the prompt and instruction used in this paper. For more details about prompting, the reader is referred to Appendix B.

In all experiments, LM outputs are generated using a greedy decoding setting (i.e. temperature = 0). We set a maximum generated token threshold of 1,024 and do not apply any frequency or presence penalties. We perform manual evaluation of the LM generations for a subset of problems, and automatic evaluations for all problems in the benchmark dataset.

### 4.1 Manual Evaluation

In the manual evaluation process, we aim to estimate the performance of GPT-3.5 and GPT-4 by manually assessing the quality of their generated instructor utterances. At each instructor dialogue turn, we manually examine each LM utterance to determine if it is an appropriate Socratic utterance at that turn. We sample a total of 17 instructor turns across 5 dialogues from the benchmark. Using the example listed in §2.1, during the second instructor turn a good-matching generated utterance example is: “How does the range function work in your loop, and what values does it generate for i?” because it is semantically close with the ground truth utterance: “Can you tell me what range(n) does?”. If the LM utterance is good but not present in our dataset, we mark it as missing to compute an overall upper bound on recall for the dataset itself. These missing alternatives can later be used to augment the

dataset. An example of a good LM utterance that is not in the dataset: “Let’s take a closer look at the loop in your code. Can you explain how the loop iterates and what it does in each iteration?”, this utterance is distinct from the first alternative response as the generated utterance gives more autonomy to the student by simply asking the student to explain the buggy portion of the code with less guidance on what to explain or look for. If the LM output is not good, it is considered a false positive (FP), which decreases the precision of the LM. An example of a poor-matching utterance in the same setting is: “Can you think of a way to modify the loop so that it starts with a different value of i?”. This utterance is generated too early before the student realizes that the loop starts with an ‘i’ value of 0. For each alternative in the benchmark dataset at that turn, we check if it is missing from the list of LM utterances. If missing, it is considered a false negative (FN), which decreases the recall of the LM. If the dataset utterance is present in the LM utterances, it is considered a true positive (TP). LM and dataset instructor utterances are matched only if they are semantic equivalent. If the LM generates two or more paraphrases of the same Socratic guidance, for the purpose of evaluation they are considered as one Socratic utterance. The precision (P), recall (R), and their harmonic mean (F1) presented in Table 2 highlight GPT-4’s superior performance over GPT-3.5 in generating relevant and diverse Socratic utterances. We emphasize GPT-3.5’s poor precision as it tends to generate many poor Socratic questions (93 FP) compared to GPT-4 (41 FP) that may contain keywords in common with a ground truth utterance but are irrelevant. In addition to the evaluation of language models, we compute the (R) for our benchmark dataset during human evaluations, obtaining a value of 70.0. This score suggests that most of the high-quality Socratic utterances generated by language models are effectively captured within the dataset by contributors, further validating the dataset’s usefulness for benchmarking purposes.

## 4.2 Qualitative Analysis

We examine the LM-generated outputs in our manual evaluation setup and observe interesting phenomena. In general, GPT-3.5 seemed to refer to example cases in the problem description to ask for the expected output in its utterances whereas GPT-4 rarely did so. Furthermore, GPT-4 has gen-

erated utterances that are good (32) but not in the dataset 3 times more than GPT-3.5 (10) indicating that GPT-4 generates more diverse (e.g. in the level of guidance) and comprehensive Socratic utterances. GPT-3.5 has generated over 96 poor utterances (FP) which is more than double that of GPT-4 (42). Below are cherry and lemon picked LM output samples given a buggy code containing a return statement within the for loop leading to an early termination. Good responses are bolded and colored in **green** while poor utterances are italicized and colored in *red*.

### ➤ Buggy code:

```
1. def removeEven(nums):
2.     newNums = []
3.     for i in nums:
4.         if i % 2 != 0:
5.             newNums.append(i)
6.         return newNums
```

### ➤ Conversation So Far:

[1] STUDENT: Help, I’m stuck.

[1] INSTRUCTOR: What should your code be doing?

[2] STUDENT: Right now, it’s only returning the first number in the list, not all of them.

[2] INSTRUCTOR: Okay, let’s look at your loop. Do you see anything that could be causing it to end early?

[3] STUDENT: No I do not. I’m iterating through the whole list of numbers.

[3] INSTRUCTOR: *(Socratic guidance)*

### ➤ Sample GPT-4 Outputs:

1. **Can you identify which part of your code is responsible for returning the final result?**
2. **What happens when the ‘return’ statement is executed inside the loop?**
3. **How does the placement of the ‘return’ statement affect the execution of your code?**
4. *How can you modify the position of the ‘return’ statement to ensure that the entire list is processed before returning the result? (direct)*

### ➤ Sample GPT-3.5 Outputs:

1. **What does the ‘return’ statement do?**

2. *How can you modify the code to return all odd numbers in the list? (early)*
3. *What happens when the condition on line 4 is true? (irrelevant)*
4. *Can you identify any issues with the loop? (repeated)*

In the example above, we observe that GPT-4 covers three distinct levels of guidance. The first utterance simply guides the student’s focus onto the erroneous line and the second utterance exerts higher level of guidance by asking the student to demonstrate their understanding of the ‘return’ keyword, and lastly the third exerts even more guidance by asking the student to explain the impact of indentation on code execution. GPT-3.5’s second utterance is illustrative of a poor utterance as it provides very little guidance and is unhelpful for the student in that conversation. Poor utterances for both LMs fall into 4 categories. The first and largest category are *irrelevant* utterances, where the SQ diverts the learner’s attention away from the actual bug and may mislead them as a consequence. GPT-3.5 has generated over 53 irrelevant utterances significantly more compared to GPT-4 (8). An example of an irrelevant utterance is the third GPT-3.5 utterance where the LM directs the focus of the learner away from the loop and why it might be terminating early and towards explaining the if statement and its body where there is no bug. The sudden shift in the goal of the conversation from discussing possible causes of the bug to explaining non-buggy code lines may mislead the learner to thinking the if statement and its body may be causing the bug when they are not. This category of utterances must be minimized by systems performing Socratic questioning. The second category are *repeated* Socratic utterances that had been asked in a prior turn or the answer to the Socratic question was given by the student in a prior turn. For example, the fourth GPT-3.5 utterance asking if the student observes any issues with the loop coming right after the student had said they don’t see anything causing the loop to end early. The third category are SQs that are *too direct* by making the bug fix pretty obvious early in the conversation. An illustrative example of this is the fourth GPT-4 utterance where it makes the bug fix obvious which is de-indenting the return statement before the student discovers the cause of the bug. These utterances lower the challenge level for students while learning and prevent stu-

dents from engaging in a discovery process and potentially lowers learning outcomes. The last category is composed of SQs uttered *too early* in the conversation, where student is not yet aware of the issue, and the Socratic utterances guide the student towards changing the code before they realize what the issue is. Take the second GPT-3.5 utterance as an example, where the LM asks the student how can they modify their code to fix the bug before the student even discovers the cause of the bug. This category of poor utterances may cause confuse learners.

### 4.3 Automatic Evaluation

Following prior work in Socratic sub-question generation (Shridhar et al., 2022), we compute the similarity between an LM utterance and a ground truth utterance in the dataset using BLEU (Papineni et al., 2002) for n-gram overlap, BERT F1 Score (Zhang et al., 2020) for semantic similarity based on the DeBERTa language model<sup>5</sup> (He et al., 2020), and Rouge-L (Lin, 2004) for n-gram overlap based on Longest Common Subsequence (LCS) between generated and reference instructor utterances. Rouge-L is included for its flexibility in evaluating text similarity and capturing overall structure and content better than BLEU-4. BERTScore is included to handle paraphrases. Given a set of  $m$  LM-generated utterances and  $n$  manually created utterances, we create a complete bipartite graph between the two sets, with a total of  $mn$  edges, where the weight of each edge is computed using one of the text similarity measures above. We then apply Edmond’s Blossom algorithm (Galil, 1986) for finding the maximum matching in this bipartite graph. This ensures that each manual utterance is matched with at most one LM utterance, effectively prohibiting semantically equivalent LM utterances from artificially increasing the evaluation measures. The number of true positives  $TP$  is computed by summing up the weights of all edges found in the optimal matching. Given that the weights are similarity scores in  $[0, 1]$ , if an LM utterance  $u$  is matched with a manual utterance  $v$  for a similarity weight of  $s(u, v)$ , the remaining weight mass of  $1 - s(u, v)$  is considered to contribute towards the total number of false positives  $FP$ . Any unmatched LM utterance is considered to contribute the maximum of 1 towards the  $FP$  total. Overall, it can be shown that this results in  $FP = m - TP$ . The number of false

<sup>5</sup><https://huggingface.co/microsoft/deberta-xlarge-mnli>

negatives is computed in an analogous way, resulting in  $FN = n - TP$ . Consequently, precision is  $P = TP/m$  and recall is  $R = TP/n$ .

The results of evaluating GPT-3.5 and GPT-4 on the entire benchmark dataset using these automated metrics are shown in Table 2. We observe that there is a correlation in terms of F1 and R between the automatic metrics and the manual metrics. However, upon manual inspection reveals that automatic evaluation metrics tend to increase when generated Socratic questions contain variable names from the buggy code input or statements from the bug description. This occurs regardless of the question’s relevance or usefulness to the student, emphasizing the importance of manual evaluation for this task.

## 5 Related Work

► **Education and Socratic Questioning.** Scaffolding is the process that enables a learner to achieve a goal through guided efforts (Wood et al., 1976). Scaffolding efforts typically focus on diversifying course content and difficulty (Saule, 2018; Dorodchi et al., 2020), however, scaffolding can also take the form of a conversation. Socratic Questioning (SQ), also referred to as guided inquiry, folds under the theory of scaffolding (Wood et al., 1976; Reiser, 2004) where a more knowledgeable person helps a learner solve a problem that is beyond their zone of proximal development (Quintana et al., 2004; Vygotsky, 2012) by interjecting with questions to guide the student towards a solution. Wood (1994) analyzed conversations in a math classroom and proposed two distinct types of questioning. The first is *funneling*, which aims to guide a learner using a set of questions toward the solution. The second is *focusing*, which draws a learner’s attention to important aspects of a problem (Wood, 1994). *Focusing* questions can also probe a student to reflect and articulate their own thinking (National Council of Teachers of Mathematics, 2014; Alic et al., 2022).

Students can complete a programming exercise but still struggle to explain their own program (Lehtinen et al., 2021). To remedy this, Tamang et al. (2021) showed that using the Socratic method to guide students in explaining their code is effective at inducing learning gains in code comprehension tasks. To the best of our knowledge, the impact of Socratic questioning on learning outcomes when guiding student debugging has not been explored yet. In this work, we create So-

cratic conversations between an instructor and a student where the instructor aims at guiding the student towards fixing a bug in their code using both *funneling* and *focusing* questions while limiting instructor utterances that provide information or facts related to fixing the bug.

► **AI for Programming Education and Dialogue Tutoring Systems.** Prior work in AI for programming education includes intelligent tutoring systems (ITS) and learning support systems for programming courses. Learning support systems provide automated feedback on student code submissions and generate programming exercises, unit tests, and code explanations (Wu et al., 2021; Sarsa et al., 2022). Most ITS models rely on methods predating recent developments in large language models (Crow et al., 2018; Mousavinasab et al., 2021), such as action-rules, Bayesian networks, and Fuzzy rules-based systems (Costello, 2012; Butz et al., 2006; Chrysafiadi and Virvou, 2012). Some work has been done in building automatic Socratic tutoring systems, but the Socratic utterances are predefined and manually specified for each exercise, limiting their generalizability (Al-shaikh et al., 2020b,a). Existing systems do not propose learning-centered conversational assistants that can generalize to unseen programming problems or focus on using Socratic questions as the main form of interaction with the learner. Automatically scaffolding learning content is important for personalized learning. Research by Kim et al. (2018) has shown that computer-based scaffolding techniques, such as hints, have a moderate impact on student learning in STEM education, paving the way for technologies to assist in the learning process. One such approach, proposed by Shridhar et al. (2022), involves automatically generating funneling Socratic sub-questions for a given math word problem using a T5 language model (Raffel et al., 2020) fine-tuned with reinforcement learning. Similarly, Tyen et al. (2022) introduce a re-ranking-based decoding strategy for language models, which adjusts the difficulty level of a chatbot to meet the needs of learners studying English as a new language.

► **Hint Generation.** With the goal of assisting students with programming exercises, recent work has proposed an array of techniques to automatically generate hints to guide novices by providing instant and relevant feedback to correct programming mistakes and advance through ex-

ercises (McBroom et al., 2021). Automated hint generation systems use various approaches including extracting common bugs and scaling up instructor feedback to the common bugs (Lee et al., 2018), extracting patterns from peer data (Iii et al., 2014; Lazar et al., 2017), and generating custom solution paths (Rivers and Koedinger, 2017) which typically generalize to unseen code states within an exercise. A super-bug is where a student incorrectly "attributes foresightedness" to the written program where the program executes beyond the information given or the student assumes there is more functionality in the written code than what was written (Pea, 1986). Fragile knowledge is broken down into four categories: missing knowledge where necessary knowledge has not been acquired, inert knowledge where the student has acquired the necessary knowledge but fails to retrieve it, misplaced knowledge where knowledge is used in the wrong context, and conglomerated knowledge where knowledge is misused by combining two or more known structures incorrectly (Perkins and Martin, 1986). Bugs caused by knowledge breakdowns where a student has a misconception are the most time-consuming to fix. For a survey on student misconceptions when learning programming the reader is referred to (Qian and Lehman, 2017).

► **Tutoring Dialogue Corpora.** Prior work in curating corpora of tutoring dialogues between an instructor and a learner includes the CIMA corpus focused on tutoring English speakers to learn Italian (Stasaski et al., 2020). Similarly, for learning English, the Teacher-Student Chatroom Corpus (TSCC), curates up to 260 chatroom dialogues between an experienced teacher and an English learner (Caines et al., 2020, 2022). TSCC was annotated according to the Self-Evaluation of Teacher Talk framework (Walsh, 2006) which includes: Enquiry (where the learner asks a question), Display Question (a question to which the teacher knows the answer), Form-focused feedback, and Instruction. Demszky et al. (2021) release a conversational corpus between math teachers and learners composed of 2,246 utterance exchanges along with annotations on teacher uptake where the teacher builds on what the student has said such as acknowledgment and rephrasing. Chen et al. (2011) examine computer science tutoring conversations and classify tutor utterances into 4 categories: The first category is *Direct Procedural Instructions*, in which the tutor directly tells the student what task

to perform. The second category is *Direct Declarative Instruction*, where the tutor provides facts about the domain or problem. The third category is *Prompts*, in which the tutor attempts to elicit a contribution from the student, and the last category is *Feedback* where the tutor affirms or rejects a step a student has completed. One interesting phenomenon observed in the corpus is tutors using analogies to communicate data structures concepts such as using Legos as an analogy to explain stacks (Alizadeh et al., 2015). Prior work focuses on building corpora of tutoring dialogues that contain instructor teaching, and tutorials. There seems to be limited work on building corpora where the instructor’s role is limited to guiding the student to discover the bug and any necessary knowledge to fix it on their own using Socratic questioning.

► **Evaluating the Educational Abilities of Language Models.** Tack and Piech (2022) propose using pairwise comparison tests to compare generated responses by BlenderBot (Roller et al., 2021) and GPT-3 (Brown et al., 2020) and find that both language models perform significantly worse than real teachers on understanding a student, helping a student, and speaking like a teacher on the TSCC (Caines et al., 2020, 2022), and the Uptake (Demszky et al., 2021) corpora which focus on English and Mathematics tutoring respectively.

## 6 Conclusion & Limitations

This paper presents a dataset of expert-curated Socratic conversations where instructors assist novice programmers in fixing buggy solutions to simple computational problems. The dataset serves as a benchmark for evaluating the Socratic debugging capabilities of LMs. While GPT-4 outperforms GPT-3.5, its precision, and recall remain below human expert levels (70.0), highlighting the need for further research. We find that GPT-family language models may generate repetitive and irrelevant Socratic utterances that could mislead learners. The utterances may also appear too early in the conversation, causing confusion, and can be overly direct, potentially diminishing learning outcomes. Study limitations include: The automatic metrics are limited in capturing the correctness, helpfulness, and relevance of a Socratic utterance, and the benchmark dataset may not represent all common novice misconceptions. Moreover, the manual evaluation is limited to 5 dialogues and could be expanded, but this process is highly time-consuming.

## Acknowledgements

We would like to thank Sandra Wiktor, Anusha Reddy, Justin Smith, and Qiong Cheng for all their time and effort in contributing dialogues to the benchmark dataset. We also acknowledge Ilan Aktanova and Frank Garcia for their contributions to the programming exercises used in the dataset and Abraham Sanders for his discussions related to dialogue system evaluations. This research was partly supported by the United States Air Force (USAF) under Contract No. FA8750-21-C-0075. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the USAF.

## References

- Sterling Alic, Dorottya Demszky, Zid Mancenido, Jing Liu, Heather Hill, and Dan Jurafsky. 2022. Computationally identifying funneling and focusing questions in classroom discourse. *BEA 2022*, page 224.
- Mehrdad Alizadeh, Barbara Di Eugenio, Rachel Harsley, Nick Green, Davide Fossati, and Omar AlZoubi. 2015. A study of analogy in computer science tutorial dialogues. *Trees*, 53(19.2):1–6.
- Zeyad Alshaikh, Lasagn Tamang, and Vasile Rus. 2020a. A socratic tutor for source code comprehension. In *Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part II 21*, pages 15–19. Springer.
- Zeyad Alshaikh, Lasang Jimba Tamang, and Vasile Rus. 2020b. Experiments with a socratic intelligent tutoring system for source code understanding. In *The Thirty-Third International Florida Artificial Intelligence Research Society Conference (FLAIRS-32)*.
- Stella Biderman and Edward Raff. 2022. **Fooling moss detection with pretrained language models**. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, CIKM '22*, page 2933–2943, New York, NY, USA. Association for Computing Machinery.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Cory J Butz, Shan Hua, and R Brien Maguire. 2006. A web-based bayesian intelligent tutoring system for computer programming. *Web Intelligence and Agent Systems: An International Journal*, 4(1):77–97.
- Andrew Caines, Helen Yannakoudakis, Helen Allen, Pascual Pérez-Paredes, Bill Byrne, and Paula Buttery. 2022. **The teacher-student chatroom corpus version 2: more lessons, new annotation, automatic detection of sequence shifts**. In *Proceedings of the 11th Workshop on NLP for Computer Assisted Language Learning*, pages 23–35, Louvain-la-Neuve, Belgium. LiU Electronic Press.
- Andrew Caines, Helen Yannakoudakis, Helena Edmondson, Helen Allen, Pascual Pérez-Paredes, Bill Byrne, and Paula Buttery. 2020. **The teacher-student chatroom corpus**. In *Proceedings of the 9th Workshop on NLP for Computer Assisted Language Learning*, pages 10–20, Gothenburg, Sweden. LiU Electronic Press.
- Tracy Camp, W Richards Adrion, Betsy Bizot, Susan Davidson, Mary Hall, Susanne Hambrusch, Ellen Walker, and Stuart Zweben. 2017. Generation cs: the growth of computer science. *ACM Inroads*, 8(2):44–50.
- Lin Chen, Barbara Di Eugenio, Davide Fossati, Stellan Ohlsson, and David Cosejo. 2011. **Exploring effective dialogue act sequences in one-on-one computer science tutoring dialogues**. In *Proceedings of the Sixth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 65–75, Portland, Oregon. Association for Computational Linguistics.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde, Jared Kaplan, Harri Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.
- Konstantina Chrysafiadi and Maria Virvou. 2012. Evaluating the integration of fuzzy logic into the student model of a web-based learning environment. *Expert systems with applications*, 39(18):13127–13134.
- Robert Costello. 2012. *Adaptive intelligent personalised learning (aipl) environment*. Ph.D. thesis.
- Tyne Crow, Andrew Luxton-Reilly, and Burkhard Wuen-sche. 2018. Intelligent tutoring systems for programming education: a systematic review. In *Proceedings of the 20th Australasian Computing Education Conference*, pages 53–62.
- Dorottya Demszky, Jing Liu, Zid Mancenido, Julie Cohen, Heather Hill, Dan Jurafsky, and Tatsunori Hashimoto. 2021. **Measuring conversational uptake: A case study on student-teacher interactions**. 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 1638–1653, Online. Association for Computational Linguistics.
- Mohsen M Dorodchi, Nasrin Dehbozorgi, Aileen Benedict, Erfan Al-Hossami, and Alexandria Benedict. 2020. Scaffolding a team-based active learning

- course to engage students: A multidimensional approach. In *2020 ASEE Virtual Annual Conference Content Access*.
- James Finnie-Ansley, Paul Denny, Brett A. Becker, Andrew Luxton-Reilly, and James Prather. 2022. [The robots are coming: Exploring the implications of openai codex on introductory programming](#). In *Australasian Computing Education Conference, ACE '22*, page 10–19, New York, NY, USA. Association for Computing Machinery.
- Zvi Galil. 1986. Efficient algorithms for finding maximum matching in graphs. *ACM Computing Surveys (CSUR)*, 18(1):23–38.
- Alex Gerdes, Bastiaan Heeren, Johan Jeuring, and L Thomas Van Binsbergen. 2017. Ask-elle: an adaptable programming tutor for haskell giving automated feedback. *International Journal of Artificial Intelligence in Education*, 27(1):65–100.
- Prakhar Gupta, Shikib Mehri, Tiancheng Zhao, Amy Pavel, Maxine Eskenazi, and Jeffrey P Bigham. 2019. Investigating evaluation of open-domain dialogue systems with human generated multiple references. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 379–391.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.
- Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, and Jacob Steinhardt. 2021. Measuring coding challenge competence with apps. *NeurIPS*.
- Sebastian Hobert. 2019. [Say Hello to ‘Coding Tutor’! Design and Evaluation of a Chatbot-based Learning System Supporting Students to Learn to Program](#). *ICIS 2019 Proceedings*.
- Barry Peddycord Iii, Andrew Hicks, and Tiffany Barnes. 2014. Generating hints for programming problems using intermediate output. In *Educational Data Mining 2014*. Citeseer.
- Johan Jeuring, L. Thomas van Binsbergen, Alex Gerdes, and Bastiaan Heeren. 2014. [Model solutions and properties for diagnosing student programs in ask-elle](#). In *Proceedings of the Computer Science Education Research Conference, CSERC '14*, page 31–40, New York, NY, USA. Association for Computing Machinery.
- Nam Ju Kim, Brian R Belland, and Andrew E Walker. 2018. Effectiveness of computer-based scaffolding in the context of problem-based learning for STEM education: Bayesian meta-analysis. *Educational Psychology Review*, 30:397–429.
- Timotej Lazar, Martin Možina, and Ivan Bratko. 2017. Automatic extraction of ast patterns for debugging student programs. In *Artificial Intelligence in Education: 18th International Conference, AIED 2017, Wuhan, China, June 28–July 1, 2017, Proceedings 18*, pages 162–174. Springer.
- Victor CS Lee, Yuen-Tak Yu, Chung Man Tang, Tak-Lam Wong, and Chung Keung Poon. 2018. Vida: A virtual debugging advisor for supporting learning in computer programming courses. *Journal of Computer Assisted Learning*, 34(3):243–258.
- Teemu Lehtinen, Aleksi Lukkarinen, and Lassi Haaranen. 2021. Students struggle to explain their own program code. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*, pages 206–212.
- 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.
- Mary Lou Maher, Celine Latulipe, Heather Lipford, and Audrey Rorrer. 2015. Flipped classroom strategies for cs education. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, pages 218–223.
- Jessica McBroom, Irena Koprinska, and Kalina Yacef. 2021. A survey of automated programming hint generation: The hints framework. *ACM Computing Surveys (CSUR)*, 54(8):1–27.
- Diba Mirza, Phillip T Conrad, Christian Lloyd, Ziad Matni, and Arthur Gatin. 2019. Undergraduate teaching assistants in computer science: a systematic literature review. In *Proceedings of the 2019 ACM Conference on International Computing Education Research*, pages 31–40.
- Elham Mousavinasab, Nahid Zarifshanaiey, Sharareh R. Niakan Kalhori, Mahnaz Rakhshan, Leila Keikha, and Marjan Ghazi Saeedi. 2021. Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods. *Interactive Learning Environments*, 29(1):142–163.
- National Council of Teachers of Mathematics. 2014. *Principles to actions: Ensuring mathematical success for all*. NCTM, National Council of Teachers of Mathematics, Reston, VA.
- OpenAI. 2022. [Introducing chatgpt](#).
- OpenAI. 2023. [Gpt-4 technical report](#). *arXiv*.
- 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.

- Roy D Pea. 1986. Language-independent conceptual “bugs” in novice programming. *Journal of educational computing research*, 2(1):25–36.
- David N Perkins and Fay Martin. 1986. Fragile knowledge and neglected strategies in novice programmers. In *Papers presented at the first workshop on empirical studies of programmers on Empirical studies of programmers*, pages 213–229.
- Yizhou Qian and James Lehman. 2017. Students’ misconceptions and other difficulties in introductory programming: A literature review. *ACM Transactions on Computing Education (TOCE)*, 18(1):1–24.
- Chris Quintana, Brian J. Reiser, Elizabeth A. Davis, Joseph Krajcik, Eric Fretz, Ravit Golan Duncan, Eleni Kyza, Daniel Edelson, and Elliot Soloway. 2004. A scaffolding design framework for software to support science inquiry. *Journal of the Learning Sciences*, 13(3):337–386.
- 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*, 21:1–67.
- Brian J Reiser. 2004. Scaffolding complex learning: The mechanisms of structuring and problematizing student work. *The Journal of the Learning sciences*, 13(3):273–304.
- Kelly Rivers and Kenneth R Koedinger. 2017. Data-driven hint generation in vast solution spaces: a self-improving python programming tutor. *International Journal of Artificial Intelligence in Education*, 27:37–64.
- Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.
- Sami Sarsa, Paul Denny, Arto Hellas, and Juho Leinonen. 2022. Automatic generation of programming exercises and code explanations using large language models. In *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 1, ICER ’22*, page 27–43, New York, NY, USA. Association for Computing Machinery.
- Erik Saule. 2018. Experiences on teaching parallel and distributed computing for undergraduates. In *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 361–368.
- Kumar Shridhar, Jakub Macina, Mennatallah El-Assady, Tanmay Sinha, Manu Kapur, and Mrinmaya Sachan. 2022. Automatic generation of socratic subquestions for teaching math word problems. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4136–4149, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Katherine Stasaski, Kimberly Kao, and Marti A Hearst. 2020. Cima: A large open access dialogue dataset for tutoring. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–64.
- Anaïs Tack and Chris Piech. 2022. The AI teacher test: Measuring the pedagogical ability of blender and GPT-3 in educational dialogues. In *Proceedings of the 15th International Conference on Educational Data Mining*, pages 522–529, Durham, United Kingdom. International Educational Data Mining Society.
- Lasang Jimba Tamang, Zeyad Alshaikh, Nisrine Ait Khayi, Priti Oli, and Vasile Rus. 2021. A comparative study of free self-explanations and socratic tutoring explanations for source code comprehension. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education, SIGCSE ’21*, page 219–225, New York, NY, USA. Association for Computing Machinery.
- Gladys Tyen, Mark Brenchley, Andrew Caines, and Paula Buttery. 2022. Towards an open-domain chatbot for language practice. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 234–249, Seattle, Washington. Association for Computational Linguistics.
- Lev S Vygotsky. 2012. *Thought and language*. MIT press.
- Steve Walsh. 2006. *Investigating classroom discourse*. Routledge.
- Judith D Wilson. 1987. A socratic approach to helping novice programmers debug programs. *ACM SIGCSE Bulletin*, 19(1):179–182.
- David Wood, Jerome S Bruner, and Gail Ross. 1976. The role of tutoring in problem solving. *Child Psychology & Psychiatry & Allied Disciplines*.
- Terry Wood. 1994. Patterns of interaction and the culture of mathematics classrooms. In *Cultural perspectives on the mathematics classroom*, pages 149–168. Springer.
- Mike Wu, Noah Goodman, Chris Piech, and Chelsea Finn. 2021. Prototransformer: A meta-learning approach to providing student feedback. *arXiv preprint arXiv:2107.14035*.



Aman Yadav, Sarah Gretter, Susanne Hambrusch, and Phil Sands. 2016. Expanding computer science education in schools: understanding teacher experiences and challenges. *Computer Science Education*, 26(4):235–254.

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 Data Contribution Web Application

We developed a 7-page data contribution web application tool using the Streamlit Python library<sup>6</sup> to collect dialogues and code snapshots. The application loads a repository of programming problems and bugs from a Google Spreadsheet using the Google Spreadsheet API through the gsheetsdb Python library<sup>7</sup>. The web app consists of the following pages:

- **Getting Started:** This page (Figure 1) orients the users on the task and provides a link to the guidelines document.
- **Browse Bugs:** Contributors browse and select a bug (Figure 2) to create a Socratic dialogue for.
- **4 Data Contribution Pages:** These pages contain a code editor and a chat area (Figure 3) where contributors create an initial conversation and up to 3 conversational threads.
- **Review and Submit:** This page (Figure 7) allows contributors to review their work and submit the exported dialogues for review.

During the data contribution process, contributors can add main and alternative utterances, undo added utterances or code snapshots, and edit the chat history text area and code in the code editor. When the contributor edits the code in the Code Editor, they can choose to compile and run the code within the web application and they can also add a code snapshot to the chat history by clicking the "Add Code to Chat History" button (Figure 4). Once the bug has been fixed, the contributor compiles and runs the code in the Code Editor, as demonstrated in Figure 5. Contributors can then use the import and export buttons shown in Figure 6 to save their work. The export button generates a standardized form of the dialogue and code states, while the import button allows contributors to load previously exported dialogues back into the tool. After completing their data contribution, contributors submit the exported dialogues for review.

## Welcome to the Socratic Debugging Project!

The aim of this project is to develop AI agents that help novice programmers debug their code through Socratic dialogue. Our first goal is to create a dataset of Socratic dialogues that can be used to train and evaluate such AI agents.

### What is Socratic Dialogue? 🤔

[Socratic dialogue](#) is named after the ancient Greek philosopher Socrates, who is known for using a method of questioning in which an expert guides a novice towards answering a question or solving a problem on their own.

### Contributing ❤️

We welcome annotations from people with good Python programming skills who are interested in helping create a dataset of Socratic dialogues for learning to code. In each dialogue, an Instructor helps a Student fix buggy implementations of simple computational problems. If you are interested in contributing, first familiarize yourself with the annotation guidelines that you can find [here](#), then follow the annotation process outlined below.

### Annotation Process Overview 📄

1. Start by browsing bugs in the [Browse Bugs](#) page. To access that page, you can either navigate using the sidebar to the left or by clicking on the "Next >" button. For each bug you will see the programming exercise, the bug, a bug description, and one or more bug fixes.
2. Once you find a bug that you would like to annotate, click on the "Annotate" button that will take you to the annotation tool.
3. Annotate a complete Socratic dialogue for that bug. Then write up to 3 conversational threads based on that dialogue.
4. When you are done, click on the "Save & Export Data" button to save your annotations into local text files. Submit the text files through the [Review & Submit](#) page that you can find in the sidebar to the left.

Figure 1: Screenshot of the web application's Getting Started page where contributors get familiarized with the task and go through the guidelines document.

<sup>6</sup><https://streamlit.io/>

<sup>7</sup><https://github.com/betodealmeida/gsheets-db-api>

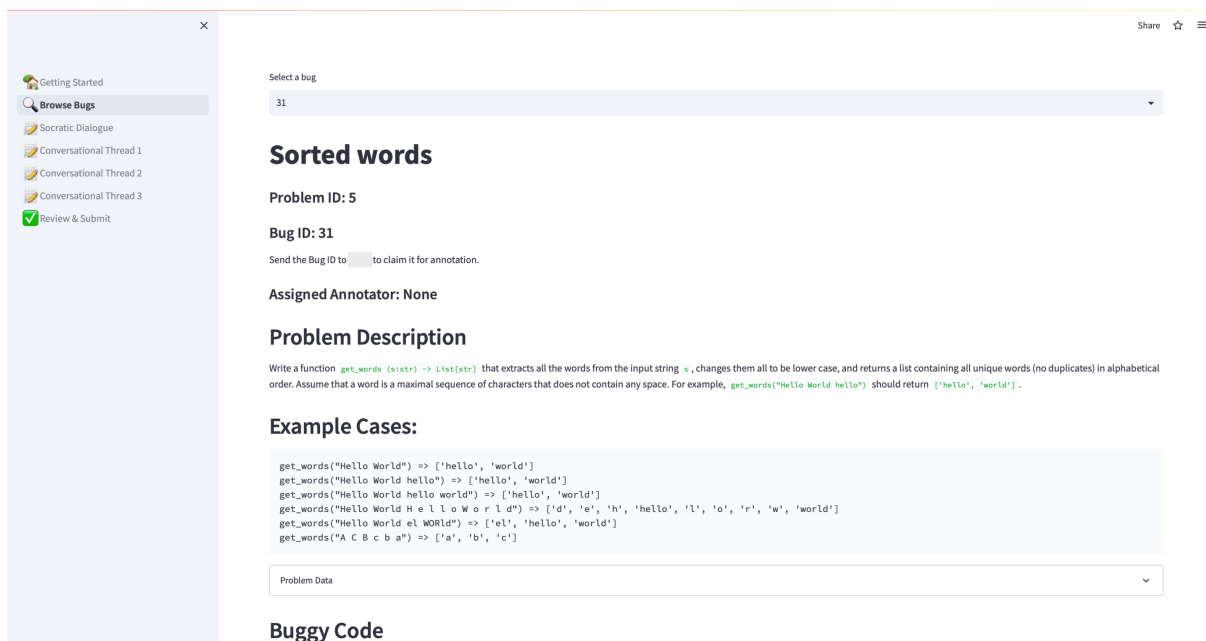


Figure 2: Screenshot of the interface. Contributors first browse a repository of bugs created from a set of programming problems. Each bug is displayed with the problem description, test cases, a buggy code, the bug description, and bug fixes. Contributors select a bug to create a dialogue for.

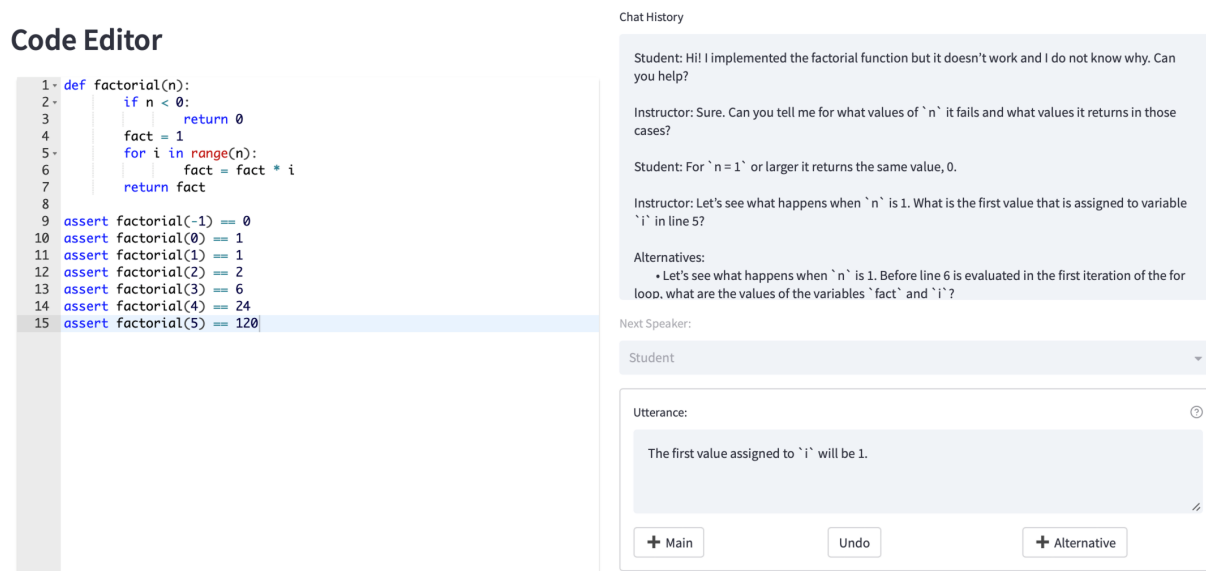


Figure 3: Screenshot of the tool used to collect dialogues and code snapshots. Contributors are able to add a main utterance, an alternative utterance, and undo an adding utterance or a code snapshot. Additionally, the chat history text area is editable.

## Code Editor

```
1 def factorial(n):
2     if n < 0:
3         return 0
4     fact = 1
5     for i in range(1, n+1):
6         fact = fact * i
7     return fact
8
9 assert factorial(-1) == 0
10 assert factorial(0) == 1
11 assert factorial(1) == 1
12 assert factorial(2) == 2
13 assert factorial(3) == 6
14 assert factorial(4) == 24
15 assert factorial(5) == 120
```

Run Code & Print Output

+ Add Code to Chat History

## Chat History

• Let's see what happens when `n` is 2. Before line 4 is evaluated in the first iteration of the for loop, what are the values of the variables `fact` and `i`?

- Let's see what happens when `n` is 1. Can you insert a new line between lines 5 and 6 that prints the values of the variables `fact` and `i`?
- Let's see what happens when `n` is 1. What does `range(n)` do when `n` is 1?
- Can you tell me what `range(n)` does?

Student: I don't know, how can I verify that?

Instructor: Can you edit the code to print the value of `i` at each iteration of the for loop?

### Alternatives:

- Can you look in the Python documentation to see what is the first value computed by range, when used with only one argument?
- Let's consider this mathematically, `fact` is assigned the value of 1 on line 4. `fact` is multiplied by all values of `i` in a range. What value would `i` need to be for `fact` to be equal to 0 after the for loop?
- Let's try it out on the terminal. Open the Python terminal using the `python` command. Then, type in a for loop similar to yours with `n` being 2. Then, in your for loop body, add in a print statement that prints `i`. What do you observe?
- Let's open the debugger. Step through your code until you reach line 6 for the first time. What do you notice about the value of `i`?

Student: Sure ... Aaah, I see, the first value is 0, not 1!

### Student Code:

```
1. def factorial(n):
2.     if n < 0:
3.         return 0
4.     fact = 1
5.     for i in range(1, n+1):
6.         fact = fact * i
7.         return fact
```

Next Speaker:

Teacher

Figure 4: Screenshot of the tool adding a code snapshot by clicking the Add Code to Chat History button.

## Code Editor

```
1 def factorial(n):
2     if n < 0:
3         return 0
4     fact = 1
5     for i in range(1, n+1):
6         fact = fact * i
7     return fact
8
9 assert factorial(-1) == 0
10 assert factorial(0) == 1
11 assert factorial(1) == 1
12 assert factorial(2) == 2
13 assert factorial(3) == 6
14 assert factorial(4) == 24
15 assert factorial(5) == 120
```

Run Code & Print Output

+ Add Code to Chat History

## Program Output

Program passes all unit tests!

Figure 5: Screenshot of the tool compiling and running the code in the Code Editor after the bug has been fixed.

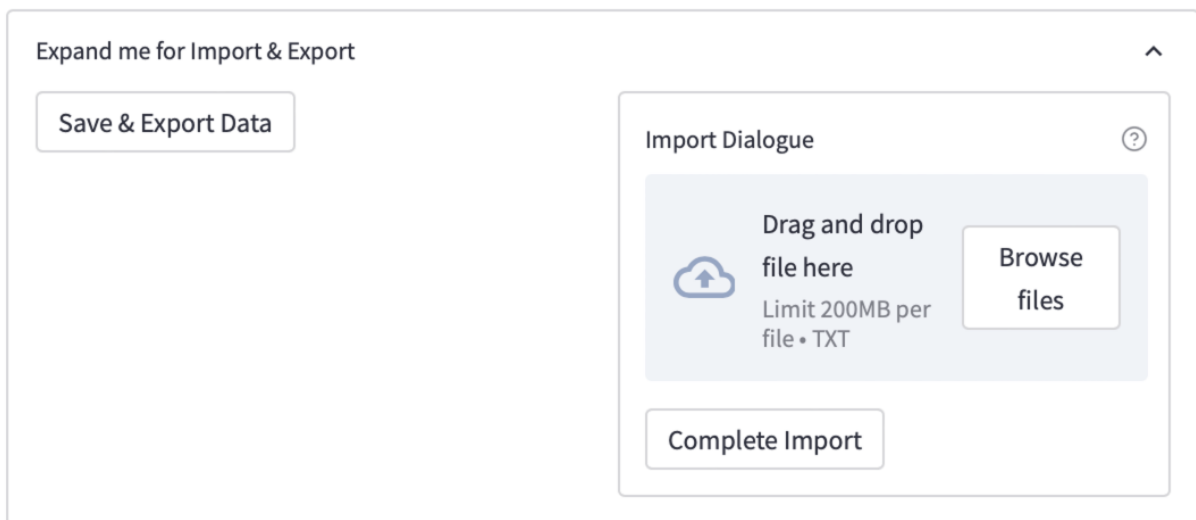


Figure 6: Screenshot of the tool’s import and export buttons. Upon completing a dialogue contributors use the export button to export the dialogue and code states into a standardized form. Additionally, contributors can import any dialogue exported from this tool using the import button.

## Review & Submit

### Review Your Dialogue

Please review your annotations carefully. If you are satisfied with your annotation, you can submit it. If you would like to make changes, you can directly edit the text file that you downloaded in the previous step. You can also use the annotation tool to make changes to your annotation.

### Common Pitfalls

1. Misindented Python code.
2. Alternative utterances are paraphrased of the main utterance.
3. The main Socratic utterance provides stronger guidance to the user than its alternatives.

For more information visit the [annotation guidelines](#).

### Check List

1. Make sure that you have downloaded all text files using the annotation tool. The files should be named as follows:
  - `*_socratic_dialogue.txt` : The main conversation between the student and the instructor.
  - `*_conversational_thread_1.txt` : The first conversational thread from the main conversation.
  - `*_conversational_thread_2.txt` : The second conversational thread from the main conversation.
  - `*_conversational_thread_3.txt` : The third conversational thread from the main conversation.
2. Make sure that you have reviewed the text files carefully and edit the text files directly.
  - Avoid the common pitfalls listed on page 11 in the [annotation guidelines](#)
  - Make sure the dialogue is coherent and the responses are appropriate.
  - Ensure that there are no typos or grammatical errors.

## Submit

Now that you have downloaded the text file using the annotation tool and reviewed it carefully, you are ready to submit!

### How to submit ?

Thank you for your contribution! Please fill out the Google Form below to submit your annotation. Note that each form submission will be reviewed by a member of our team before being added to the dataset.

Figure 7: Screenshot of the web application’s Review & Submit page where contributors are instructed to review their data contribution and submit their exported version.

## B Language Model Prompt

This section describes the prompt template that was used for language models in this paper. `{{text}}` denotes a data point from the benchmark dataset. The steering prompt was adapted from the GPT-4 blog post<sup>8</sup>. The ‘1.’ is added at the end of the instruction to prompt the language model to generate an itemized list of utterances that can then be parsed.

*Steering Prompt:*

You are a tutor that always responds in the Socratic style. You *never* give the student the answer, but always try to ask just the right question to help them learn to think for themselves. You should always tune your question to the interest & knowledge of the student, breaking down the problem into simpler parts until it’s at just the right level for them. Socratic utterances are utterances that guide the user and do not give them the solution directly. In each of your responses, provide a comprehensive list of Socratic responses that you can give to the user to help them solve the problem on their own, based on the conversation so far.

*Prompt:*

```
<problem>
{{Problem Description}}
</problem>
<bug_code>
{{Buggy Code}}
</bug_code>
<bug_desc>
{{Bug Description}}
</bug_desc>
<bug_fixes>
{{Bug Fixes}}
</bug_fixes>
<unit_tests>
{{Unit Tests}}
</unit_tests>
```

User: `{{User Turn 1}}`

Assistant: `{{Assistant Turn 1}}`

...

User: `{{User Turn N}}`

`<code>`

`{{Code State at Turn N}}`<sup>a</sup>

`</code>`

Respond to the user with all possible distinct Socratic utterances that guide the user to discover and fix the bug described between ‘<bug\_desc>’ and ‘</bug\_desc>’. Student code is written between ‘<code>’ and ‘</code>’ throughout the conversation. Utterances that have the same meaning but different words are considered duplicates. Assume that the student has run the test cases.

1.

<sup>a</sup>Included only if turn N has a code state.

<sup>8</sup><https://openai.com/research/gpt-4>

# Beyond Black Box AI-Generated Plagiarism Detection: From Sentence to Document Level

**Mujahid Ali Quidwai**  
New York University  
maq4265@nyu.edu

**Chunhui Li**  
Columbia University  
cl4282@columbia.edu

**Parijat Dube**  
IBM Research  
pdube@us.ibm.com

## Abstract

The increasing reliance on large language models (LLMs) in academic writing has led to a rise in plagiarism. Existing AI-generated text classifiers have limited accuracy and often produce false positives. We propose a novel approach using natural language processing (NLP) techniques, offering quantifiable metrics at both sentence and document levels for easier interpretation by human evaluators. Our method employs a multi-faceted approach, generating multiple paraphrased versions of a given question and inputting them into the LLM to generate answers. By using a contrastive loss function based on cosine similarity, we match generated sentences with those from the student's response. Our approach achieves up to 94% accuracy in classifying human and AI text, providing a robust and adaptable solution for plagiarism detection in academic settings. This method improves with LLM advancements, reducing the need for new model training or re-configuration, and offers a more transparent way of evaluating and detecting AI-generated text.

## 1 Introduction

In recent years, large language models (LLMs) have demonstrated remarkable capabilities across a wide range of natural language processing (NLP) tasks, including text classification, sentiment analysis, translation, and question-answering (He et al., 2023).

These foundational models exhibit immense potential in tackling a diverse array of NLP tasks, spanning from natural language understanding (NLU) to natural language generation (NLG), and even laying the groundwork for Artificial General Intelligence (AGI) (Yang et al., 2023). In the world of advanced LLMs, ChatGPT (2023) as an AI model developed by OpenAI (2023b) has become one of the most popular and widely used models, setting new records for performance and flexibility

in many applications. According to the latest available data, ChatGPT (2023) currently has over 100 million users and the website currently generates 1 billion visitors per month (Duarte, 2023). While ChatGPT has brought numerous benefits such as it allows us to obtain information more effectively, improves people's writing skills etc., however, it has also introduced considerable risks (OpenAI, 2023a).

A major risk associated with the growing dependence on ChatGPT is the escalation of plagiarism in academic writing (Khalil and Er, 2023), which subsequently compromises the integrity and purpose of assignments and examinations. Thanks to its advanced training process and access to abundant pre-training data sets, ChatGPT is capable of resembling human-like language when provided with a prompt (Joshi et al., 2023). It even exceeds human performance in some academic writing while maintaining authenticity and richness. Furthermore, humans are unable to accurately distinguish between Human Generated Text (HGT) and Machine Generated Text (MGT), regardless of their familiarity with ChatGPT (Herbold et al., 2023). These factors present significant challenges in maintaining educational integrity and challenge the current paradigm of how teachers teach.

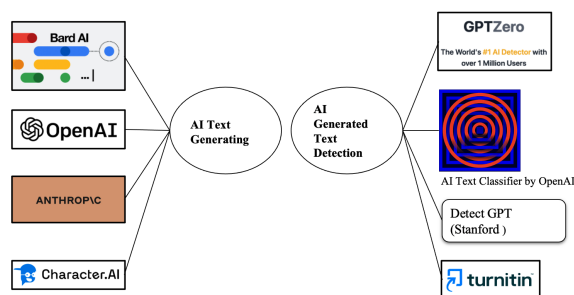


Figure 1: Popular LLMs and AI-generated text detection tools

To reduce potential plagiarism caused by the use of LLMs, researchers have developed various

AI-generated text classifiers or tools such as Log-Likelihood (Solaiman et al., 2019), RoBERTa-QA (HC3) (Guo et al., 2023), GPTZero (2023), OpenAI Classifier (OpenAI, 2023b), DetectGPT (Mitchell et al., 2023), and Turnitin (Fowler, 2023). Figure 1 lists popular LLMs used for text generation and AI-generated text detection tools.

Existing approaches for detecting text generated by language models have several limitations as highlighted in Table 1. For instance, these tools

Current Method	High False Positive	Model Retraining	Works for GPT2	Blackbox Nature
Log-Likelihood	✓	✓	✓	✓
RoBERTa-QA (HC3)	✓	✓	×	✓
OpenAI Classifier	✓	✓	×	✓
DetectGPT	✓	✓	✓	✓
Turnitin	✓	✓	×	✓

Table 1: Problems with current approaches

may rapidly become outdated due to technological advancements, such as new versions of GPT models, necessitating classifier retraining and often resulting in limited accuracy. Models trained specifically on one language model might not effectively detect text generated by a different language model (e.g., DetectGPT classifier works only on text generated using GPT2 (OpenAI, 2023b)). Additionally, some detection tools provide non-quantitative label results, and all possess a black-box nature concerning prediction accuracy. Thus the predictions made by such tools lack explainability and are challenging for human evaluators to comprehend. This issue leads to a high number of false positive punishments in academic settings (Fowler, 2023).

We propose a novel approach for detecting plagiarized text, which focuses on NLP techniques. Our approach offers more quantifiable metrics at the sentence level, allowing for easier interpretation by human evaluators and eliminating the black-box nature of existing AI text detection methods. Our approach is not limited to ChatGPT but can also be applied to other LLMs such as BardAI (Google, 2023), Character.AI (Character.AI, 2023), and so on, it also can adapt automatically as those LLMs upgrade. This adaptability helps to ensure that it

does not become outdated quickly as technology advances.

In evaluating our approach, we used the open dataset known as the ChatGPT Comparison Corpus (HC3) (Guo et al., 2023). This dataset contains 10,000 questions and their corresponding answers from both human experts and ChatGPT, covering a range of domains including open-domain, computer science, finance, medicine, law, and psychology. Our approach achieves **94%** accuracy in classifying between human answers and ChatGPT answers in the HC3 data set.

The paper is structured as follows. Section 2 contains a review of relevant literature. Our proposed end-to-end approach for AI-text detection is detailed in Section 3, where we describe the method framework. In Section 4, we present our main results from the experimental evaluation. Lastly, we summarize our findings and discuss future directions in Section 5, which serves as the conclusion.

## 2 Related Work

The field of AI-generated text detection has garnered significant interest, but only a few models and tools have achieved widespread adoption. In this section, we discuss state-of-the-art approaches, the datasets used for training their classifiers, and their limitations.

### 2.1 DetectGPT

DetectGPT (Mitchell et al., 2023) is a zero-shot machine-generated text detection method that leverages the negative curvature regions of an LLM’s log probability function. The approach does not require training a separate classifier, collecting a dataset of real or generated passages, or watermarking generated text. Despite its effectiveness, DetectGPT is limited to GPT-2 generated text, and its performance may not extend to other LLMs (Tang et al., 2023).

### 2.2 Human ChatGPT Comparison Corpus (HC3)

Guo et al. (2023) introduced the HC3 dataset, which contains tens of thousands of comparison responses from both human experts and ChatGPT (2023). They conducted comprehensive human evaluations and linguistic analyses to study the characteristics of ChatGPT’s responses, the differences and gaps from human experts, and future directions for LLMs. Furthermore, they built



three different detection systems to effectively detect whether a text is generated by ChatGPT or humans. However, this approach might still suffer from high false positive rates and it does not provide correct sentence-level comparison metrics.

### 2.3 OpenAI AI Text Classifier

The OpenAI AI Text Classifier (OpenAI, 2023b) is a fine-tuned GPT model designed to predict the likelihood of a piece of text being AI-generated. This free tool aims to foster discussions on AI literacy, but it has limitations: it requires a minimum of 1,000 characters, can mislabel AI-generated and human-written text, and can be evaded by editing AI-generated text. Additionally, it also suffers from high positive rates.

In our research, we aim to address the limitations of these existing methods by developing a novel approach for detecting plagiarized text, focusing on natural language processing techniques that provide more quantifiable metrics and eliminate the black-box nature of existing AI text detection methods

## 3 Our Method

In this section, we present our approach to effectively compare and detect plagiarism in student responses. Our method utilizes an advanced paraphrasing model, a state-of-the-art language model, and a contrastive loss function to deliver a comprehensive and transparent evaluation system. Figure 2 shows the different components of our proposed model architecture.

### 3.1 Paraphrasing Model

To simulate the variety of questions a student might pose to a large language model (LLM), we employ a paraphrasing model (refer to Figure 2). This model generates multiple paraphrased versions of a given question, accounting for the diversity in student queries and ensuring robustness in the detection process.

Using Google's T5 language model (Roberts and Raffel, 2020) an initial dataset of questions is selected and paraphrased. The T5 model is trained on a large corpus of text, which enables it to understand the context and rephrase questions that preserve the original meaning while introducing variety. This model mimics the array of questions students might ask an LLM, a crucial feature to ensure robustness in the detection process.

For example, consider the following original question Q1 from the Reddit ELI5 (HC3 dataset) (Guo et al., 2023):

Q1 What is a hacker really doing? I've always wanted to know why hackers are jamming on their keyboards and what they can possibly be doing that requires such precision and speed. Please explain like I'm five.

The paraphrasing model generates the following paraphrased versions:

P1 What do hackers actually do? I've always been curious about why they type so fast on their keyboards and what they're doing that needs such accuracy and quickness. Can you explain it to me as if I were a five-year-old?

P2 I've always wondered what hackers are truly up to when they're typing rapidly on their keyboards. What kind of tasks are they performing that demand such skill and swiftness? Please explain it in a simple way, as if I were just five years old.

P3 What is it that hackers do when they're furiously typing on their keyboards? I'm curious about the activities they're involved in that require such speed and precision. Could you break it down for me in terms a five-year-old would understand?

### 3.2 LLM Integration

Once we have the paraphrased questions, we input them into an LLM. This model, particularly the GPT-3.5-turbo from OpenAI ChatGPT (2023), is adept at generating coherent and contextually appropriate answers. Its proficiency stems from pre-training on an extensive amount of textual data, allowing it to provide accurate and relevant responses to the paraphrased questions.

For example, we input the first paraphrased question P1 into the LLM and generate the following answer:

A1 Hackers are like computer experts who solve puzzles. They use their keyboards to give commands to computers to find secret information or fix problems. They need to be fast and accurate because computers follow instructions very quickly, and one wrong command can cause mistakes. Just like playing a game, they need to be good at using their keyboards to win the computer puzzle.

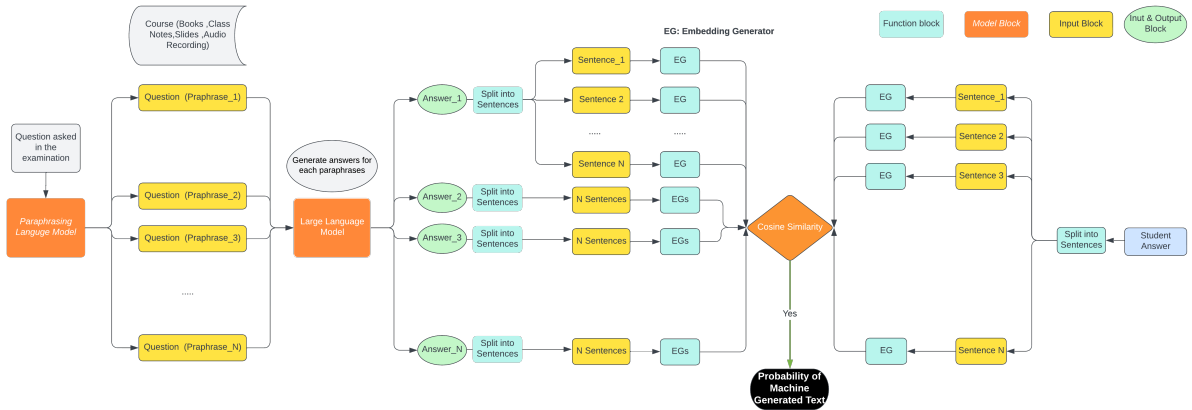


Figure 2: Model Architecture for our proposed method

We do similar generations for the other two paraphrased versions of the question Q1.

### 3.3 Evaluation Process

To facilitate a detailed comparison between the LLM-generated answers and student responses, we break down each answer into individual sentences. This granular approach enhances transparency and allows for a more in-depth evaluation of potential plagiarism.

For example, consider the LLM-generated answer A1 and a human answer H1 for question Q1 from the Reddit ELI5 dataset:

H1 I've always wanted to know why hackers are jamming on their keyboards In reality, this doesn't happen. This is done in movies to make it look dramatic and exciting. Real computer hacking involves staring at a computer screen for hours of a time, searching a lot on Google, muttering hmmm and various expletives to oneself now and then, and stroking one's hacker - beard while occasionally tapping on a few keys .", "Computers are stupid, they don't know what they are doing, they just do it. If you tell a computer to give a cake to every person that walks through the door, it will do. Hackers are the people that get extra cake by going around the building and back through the door. GLADOS however, will give you no cake .", "Hackers have a deep and complete understanding of a subject ( e.g. a machine or computer program ). They change the behavior of the subject to something that was never intended or even thought it would be possible by the creator of the subject .

We next do a pair-wise comparison between a

sentence in H1 and all the sentences in A1, A2, and A3, to identify the AI generated sentence which is most similar to H1.

### 3.4 Cosine Similarity

To compare two sentences we measure cosine similarity between the embeddings for the sentences generated using `text-embedding-ada-002`. The use of cosine similarity on sentence level contextualembeddings captures semantic and syntactic congruence between compared sentences. We use the term Human-Machine (HM) comparison for comparing sentence pairs involving a human-generated sentence and a machine-generated sentence. While Machine-Machine (MM) comparison involves comparing two machine-generated sentences.

### 3.5 Linear Discriminant Analysis

We apply Linear Discriminant Analysis (LDA) (Tharwat et al., 2017) —a supervised classification method — to categorize sentences as human- or AI-generated using cosine similarity scores. These scores and their respective category labels form our dataset, serving as independent and dependent variables, respectively. The LDA model is trained using `sklearn's LinearDiscriminantAnalysis` class. The trained model is then used to predict the probability of a sentence in the test set being AI-generated.

To optimize classification, we explore a range of threshold values from 0 to 1 in a binary system. By assigning samples in datasets HM and MM to categories 0 and 1 respectively, we can conduct the LDA analysis on these two groups of datasets. Consequently, we determine the optimal threshold for classifying human-generated text and AI-generated

text where the accuracy is maximized.

## 4 Experimental Evaluation

In our experimental evaluation, we aim to measure the accuracy of our approach in detecting similarities between human and machine-generated answers. We use the Human ChatGPT Comparison Corpus (HC3) dataset, which contains human and ChatGPT-generated answers to the same questions.

### 4.1 Dataset Preparation

For our analysis, we prepare two datasets to evaluate our model at the sentence and document levels. We use the HC3 dataset for sentence-level evaluation and then we did a summation over sentence-level cosine similarity to get the average similarity for the document. Further, to evaluate generalization performance of our model, we use GPT-wiki-intro dataset (Aaditya Bhat, 2023) for document-level evaluation and comparison with other models.

#### 4.1.1 Sentence-level Dataset: HC3

We first use the HC3 dataset, which contains questions and corresponding human and machine responses. The HC3 dataset has an additional machine response for each question, resulting in two machine-generated answers.

Next, we break down each answer for a given question into individual sentences, creating a dataset of roughly 43,000 sentence-level comparisons for machine-machine (MM) and human-machine (HM) categories. We use this dataset to compare the human response to the machine response at the sentence level, as well as compare the machine responses to each other at the sentence level using cosine similarity. Some example cosine similarity values for HM and MM categories are presented in Table 2.

HM		MM	
CS	Label	CS	Label
0.785	0	0.846	1
0.826	0	0.824	1
0.690	0	0.827	1
0.778	0	0.824	1
0.899	0	0.824	1

Table 2: Example results of cosine similarity (CS) on HM and MM sample with their corresponding categorical label (0,1)

Figure 3 shows the distribution of cosine similarity for HM and MM. For a visual representation of

the cosine similarity scores distribution, we generate a Kernel Density Estimation (KDE) plot (Chen, 2017). We also calculate the mean and standard deviation of these scores (see Table 3) for sentence level in HM and MM samples, providing insights into the data. While the mean of the two classes is significantly different, they also have high standard deviations. This dataset is to be used to train and test our LDA model at the sentence level.

Table 4 shows the threshold value used in the LDA classifier and the corresponding accuracy on the test set.

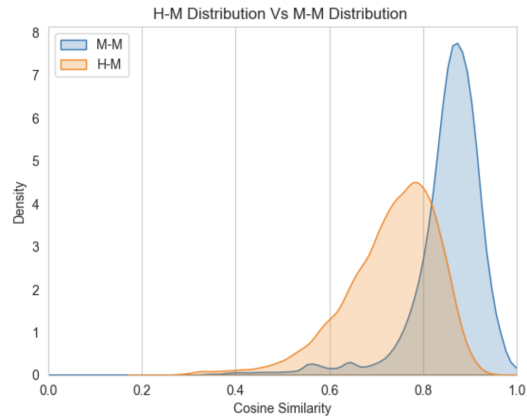


Figure 3: Distribution of cosine similarity at sentence level for HM and MM.

#### 4.1.2 Document-Level Dataset: HC3

For a comprehensive understanding, we also conduct a document-level analysis utilizing the HC3 dataset. Rather than dissecting the responses into separate sentences, this level of examination treats the entire response as a single unit.

The document-level dataset is constructed by averaging the highest cosine similarity scores from the sentence-level comparison within each response. This approach ensures that the most closely matched sentences significantly impact the document-level similarity metric, thereby emphasizing the presence of highly similar sentences in the text. This similarity value serves as the foundation for our LDA model at the document level, allowing for a macroscopic comparison of the machine and human responses.

The distribution of cosine similarity at the document level is shown in Figure 4. Table 5 provides the mean and standard deviation of cosine similarity scores for HM and MM samples in the document level dataset.

Statistic	Human-Machine (HM)	Machine-Machine (MM)
Mean	0.7309	0.8527
Standard Deviation	0.1016	0.0813

Table 3: Sentence Level Cosine Similarity Statistics

LDA Model Result	Value
Best Threshold	0.40
Accuracy	0.80

Table 4: LDA Model Results: Sentence level

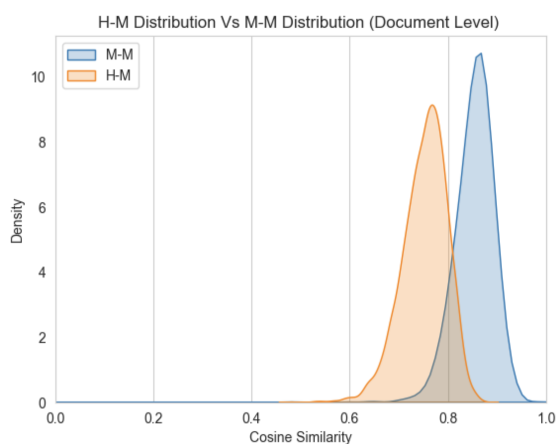


Figure 4: Distribution of cosine similarity at the Document level for HM and MM

The LDA classifier’s threshold value and the corresponding accuracy on the test set for the document-level analysis are presented in Table 6. Observe that, in contrast to sentence level statistics (Table 3), the standard deviation of the two classes under document level comparison (Table 5) are smaller thereby resulting in a more discriminant classifier.

## 4.2 Experimental Setup

Using the prepared dataset, we conduct a series of experiments to assess the performance of our proposed method in various plagiarism scenarios.

All the elements from the test set i.e., questions and corresponding student answers, including original and paraphrased questions alongside their corresponding AI-generated answers, are subsequently stored in a vector database, more specifically, Milvus (Wang et al., 2021), an open-source vector database. This step ensures efficient data management, comparison, and high-speed searching of vector data. We incorporate FastText, a module developed by Facebook (Bojanowski et al., 2017),

for vector ranking. The vectors representing paraphrased answers are ranked, creating a hierarchy of sentences based on similarity. A vector embedding generator from OpenAI aids in transforming the text into numerical form, allowing machine learning algorithms to process it. This transformation is pivotal for comparing student responses with AI-generated answers.

## 4.3 Results and Analysis

From Table 4 and Table 6 we observe that the LDA classifier works better at the document level compared to the sentence level. We next conduct the document level evaluation of our model on the GPT-wiki-intro dataset. This dataset comprises questions along with their corresponding GPT-2 generated introductions and human-written introductions from Wikipedia articles. We perform document-level analysis on the first 100 examples from the GPT-wiki-intro dataset by comparing the AI-generated introductions to the human-written introductions, as well as comparing the AI-generated introductions to each other.

Our evaluation aims to demonstrate the explainability of our tool and its ability to provide both sentence and document level analysis. By comparing our results with existing benchmarks, we highlight the advantages of our approach in detecting plagiarism more effectively and transparently.

Our model is compared with two state-of-the-art (SOTA) models: HC3 and OpenAI’s text classifier. In order to evaluate the effectiveness of using the proposed paraphrasing model, we used two versions of our model, a model without paraphrasing (A) and a model employing paraphrasing (B) on the test set. This allows us to directly assess the impact of paraphrasing on model performance.

Confusion matrices for all the models under evaluation are shown in Table 7. While derived performance metrics (F1 score, precision, and recall) are provided in Table 8. We observe no improvement in model performance with paraphrasing on this data set. We plan to investigate other potential approaches to improve model performance including varying the temperature and P value (OpenAI,

Statistic	Human-Machine (HM)	Machine-Machine (MM)
Mean	0.7343	0.8527
Standard Deviation	0.0447	0.0681

Table 5: Document Level Cosine Similarity Statistics

LDA Model Result	Value
Best Threshold	0.66
Accuracy	0.94

Table 6: LDA Model Results: Document level

2023a) of LLM used for answer generation. We also plan to study our model performance on other datasets for a robust evaluation of the value of paraphrasing.

	Predicted 0	Predicted 1
RoBERTa-QA		
Actual 0	91	9
Actual 1	77	23
OpenAI Classifier		
Actual 0	64	36
Actual 1	98	2
Our Model-A		
Actual 0	99	1
Actual 1	90	10
Our Model-B		
Actual 0	98	2
Actual 1	91	9

Table 7: Confusion matrices for SOTA models and our model tested on GPT-wiki-intro dataset. Our model performance on Class 0 is better than both RoBERTa-QA and Open AI Classifier, while on Class 1 our performance is better than RoBERTa-QA. Our Model-B uses paraphrasing.

Model	Precision	Recall	F1
RoBERTa-QA	0.91	0.54	0.68
OpenAI Classifier	0.64	0.39	0.49
<b>Our Model-A</b>	0.99	0.52	0.69
<b>Our Model-B</b>	0.98	0.52	0.68

Table 8: Document level F1 score, precision, and recall of the models. Our Model-B uses paraphrasing.

Our model provides the probability of a text be-

ing AI-generated, both at sentence and document levels, enhancing transparency for evaluators examining potential plagiarism. For each sentence in a test document - in this case, a student response - the model calculates the probability of that sentence being LLM-generated. When utilizing the Reddit ELI5 (HC3 dataset), our model contrasts the human response with the LLM response on a sentence-by-sentence basis, as demonstrated in Table 9. This added transparency makes it easier for human evaluators to interpret the results and contributes to the elimination of the black-box nature often associated with existing AI text detection methods. To summarize, our method:

- Effectively generates diverse paraphrased questions using an advanced paraphrasing model.
- Produces accurate and contextually appropriate answers with the state-of-the-art LLM.
- Provides a comprehensive and transparent sentence-level evaluation, enabling the detection of subtle instances of plagiarism that might be overlooked by traditional methods.

## 5 Conclusion

In conclusion, this research presents a novel and effective method for detecting machine-generated text in academic settings, offering a valuable contribution to the field of plagiarism detection. By leveraging a comprehensive comparison technique, our approach provides more accurate and explainable evaluations compared to existing methods. The sentence level quantifiable metrics facilitate easier interpretation for human evaluators, mitigating the black-box nature of existing AI text detection methods.

Our model is adaptable to various NLG models, including cutting-edge LLMs like BardAI and Character.AI, ensuring its relevance and effectiveness as technology continues to evolve. This adaptability makes our approach a significant asset in maintaining academic integrity in the face of rapidly advancing natural language processing technologies.

Future research directions include collecting additional unbiased datasets for evaluation and comparing the performance of our model with other detection tools. We also plan to explore the incorporation of different algorithms at the sentence level, assembling them to achieve even better performance. Moreover, we plan to employ stylometry techniques to identify each student’s unique writing style as more data from their responses are collected. This process will create a distinct signature based on the student’s writing patterns, making it increasingly easy to detect plagiarism in future submissions.

These efforts will further refine our model and contribute to the ongoing pursuit of robust, transparent, and adaptable plagiarism detection methods in academia.

## References

- Aaditya Bhat. 2023. [Gpt-wiki-intro \(revision 0e458f5\)](#).
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. [Enriching word vectors with subword information](#). *Transactions of the Association for Computational Linguistics*, 5:135–146.
- Character.AI. 2023. [Character.ai](#). <https://beta.character.ai/>. Accessed on May 15, 2023.
- ChatGPT. 2023. [Chatgpt official website](#). <https://openai.com/blog/chatgpt>. Accessed on May 15, 2023.
- Yen-Chi Chen. 2017. [A tutorial on kernel density estimation and recent advances](#).
- Fabio Duarte. 2023. [Number of chatgpt users \(2023\)](#). <https://explodingtopics.com/blog/chatgpt-users>. Accessed on May 15, 2023.
- Geoffrey A. Fowler. 2023. [We tested a new chatgpt-detector for teachers. it flagged an innocent student](#). <https://www.washingtonpost.com/technology/2023/04/01/chatgpt-cheating-detection-turnitin/>.
- Google. 2023. [Bardai](#). <https://blog.google/technology/ai/try-bard/>. Accessed on May 15, 2023.
- GPTZero. 2023. [Gptzero official website](#). <https://gptzero.me/>. Accessed on May 15, 2023.
- Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. [How close is chatgpt to human experts? comparison corpus, evaluation, and detection](#).
- Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2023. [Mgtbench: Benchmarking machine-generated text detection](#).
- Steffen Herbold, Annette Hautli-Janisz, Ute Heuer, Zlata Kikteva, and Alexander Trautsch. 2023. [Ai, write an essay for me: A large-scale comparison of human-written versus chatgpt-generated essays](#).
- Ishika Joshi, Ritvik Budhiraja, Harshal Dev, Jahnvi Kadia, M. Osama Ataullah, Sayan Mitra, Dhruv Kumar, and Harshal D. Akolekar. 2023. [Chatgpt – a blessing or a curse for undergraduate computer science students and instructors?](#)
- Mohammad Khalil and Erkan Er. 2023. [Will chatgpt get you caught? rethinking of plagiarism detection](#).
- Eric Mitchell, Yoonho Lee, Alexander Khzatsky, Christopher D. Manning, and Chelsea Finn. 2023. [Detectgpt: Zero-shot machine-generated text detection using probability curvature](#).
- OpenAI. 2023a. [Gpt-4 technical report](#).
- OpenAI. 2023b. [Openai official website](#). <https://openai.com/>. Accessed on May 15, 2023.
- Adam Roberts and Colin Raffel. 2020. [Exploring transfer learning with T5: the text-to-text transfer transformer](#). Google AI Blog. Google AI Blog.
- Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. 2019. [Release strategies and the social impacts of language models](#).
- Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. 2023. [The science of detecting llm-generated texts](#).
- Alaa Tharwat et al. 2017. [Linear discriminant analysis: A detailed tutorial](#).
- Jianguo Wang, Xiaomeng Yi, Rentong Guo, Hai Jin, Peng Xu, Shengjun Li, Xiangyu Wang, Xiangzhou Guo, Chengming Li, Xiaohai Xu, et al. 2021. [Milvus: A purpose-built vector data management system](#). In *Proceedings of the 2021 International Conference on Management of Data*, pages 2614–2627.
- Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Bing Yin, and Xia Hu. 2023. [Harnessing the power of llms in practice: A survey on chatgpt and beyond](#).

LLM Response	Human Response	Cosine Similarity
This can involve a lot of trial and error, which is why hackers might seem to be "jamming on their keyboards" as they try different approaches.	Computers are stupid , they do n't know what they are doing , they just do it.	0.8087
Overall, hacking can be a complex and technical activity that requires a lot of knowledge and skill.	Hackers have a deep and complete understanding of a subject (e.g., a machine or computer program).	0.8753
Hacking can involve a lot of typing and computer use, because hackers often use special software and programs to try to find weaknesses in a system or network.	A machine or computer program.	0.8154
This can involve a lot of trial and error, which is why hackers might seem to be "jamming on their keyboards" as they try different approaches.	GLaDOS however , will give you no cake.	0.7472
A hacker is someone who uses their computer skills to try to gain access to systems or networks without permission.	Hackers are the people that get extra cake by going around the building and back through the door.	0.8677
This can involve a lot of trial and error, which is why hackers might seem to be "jamming on their keyboards" as they try different approaches.	I 've always wanted to know why hackers are jamming on their keyboards In reality , this does n't happen.	0.8641
They might also use tools to try to guess passwords or to find ways to get around security measures.	If you tell a computer to give a cake to every person that walks through the door , it will do.	0.7735
Hacking can involve a lot of typing and computer use, because hackers often use special software and programs to try to find weaknesses in a system or network.	Real computer hacking involves staring at a computer screen for hours of a time , searching a lot on Google , muttering " hmmm " and various expletives to oneself now and then , and stroking one '.	0.8846
Hackers might do this for a variety of reasons, such as to steal information, to cause damage or disruption, or just for the challenge of it.	They change the behavior of the subject to something that was never intended or even thought it would be possible by the creator of the subject.	0.7937
Hackers might do this for a variety of reasons, such as to steal information, to cause damage or disruption, or just for the challenge of it.	This is done in movies to make it look dramatic and exciting.	0.7676

Table 9: This table depicts the sentence-level comparison of responses to Question 1 Q1, given by a human H1 and a Large Language Model A1. The cosine similarity values, derived from embeddings, represent the highest similarity between each pair of sentences in the human and LLM responses.

# Enhancing Educational Dialogues: A Reinforcement Learning Approach for Generating AI Teacher Responses

Thomas Huber and Christina Niklaus and Siegfried Handschuh

University of St. Gallen

firstname.lastname@unisg.ch

## Abstract

Reinforcement Learning remains an underutilized method of training and fine-tuning Language Models (LMs) despite recent successes. This paper presents a simple approach of fine-tuning a language model with Reinforcement Learning to achieve competitive performance on the BEA 2023 Shared Task whose goal is to automatically generate teacher responses in educational dialogues. We utilized the novel NLPO algorithm that masks out tokens during generation to direct the model towards generations that maximize a reward function. We show results for both the t5-base model with 220 million parameters from the HuggingFace repository submitted to the leaderboard that, despite its comparatively small size, has achieved a good performance on both test and dev set, as well as GPT-2 with 124 million parameters. The presented results show that despite maximizing only one of the metrics used in the evaluation as a reward function our model scores highly in the other metrics as well.

## 1 Introduction

Controlling the output of Language Models is a challenging problem in the field of Natural Language Processing (NLP). Recently Reinforcement Learning (RL) has successfully been applied to the training and fine-tuning of Language Models. ChatGPT, based on InstructGPT (Ouyang et al., 2022a), makes use of Reinforcement Learning. Ramamurthy et al. (2023) have proposed the GRUE (General Reinforced-language Understanding Evaluation) benchmark that consists of a variety of different tasks, supervised by different Reward Functions to measure the quality of the trained models. The reported results on a variety show good results on a variety of tasks. Despite recent advances in applying RL to the training and fine-tuning of LMs and their wide applicability to different tasks and benchmarks this approach is still not widely applied.

In this paper we make use of Reinforcement Learning-based fine-tuning to tackle the BEA 2023 Shared Task (Tack et al., 2023). The goal of the task is the generation of teacher-like responses in an educational dialogue setting between a student and a teacher. This necessitates that the language model can mimic the tone and overall quality of the teacher response. We have employed an approach that pushes the generations of the model in the right direction through the use of BERTScore as a reward function and using Reinforcement Learning as our training strategy.

Our model submission to the leaderboard is the implementation of the T5 model (Raffel et al., 2020) in the HuggingFace repository, t5-base with 220 million parameters. As the goal is to generate a response given an input dialogue we have chosen a sequence-to-sequence model. We follow the findings of Ramamurthy et al. (2023) who suggest that a small model with a high-quality reward function can match or outperform models with magnitudes of more parameters. For the training process we use the dialogue preceding the final teacher response as input and the final teacher response as the reference text. We achieve an average rank across all metrics of 5.38, out of 10 submissions, placing overall in seventh place on the leaderboard. For the DialogRPT maximum weighted ensemble metric our model achieves first place on the test set. We additionally present results for an autoregressive model. The chosen model is the base GPT-2 model from the HuggingFace repository with 124 million parameters. The autoregressive model outperforms our submitted model despite its smaller size in terms of parameters, suggesting that this model architecture may be more suitable for this task.

## 2 Related Work

Ramamurthy et al. (2023) present results showing that Reinforcement Learning can be applied



Tokenizer	Min	Max	Avg.
t5-base	201	9	99.17
gpt2	223	11	100.03

Table 1: Lengths of the training samples. Values are measured in tokens.

successfully in various NLP settings, including on the DailyDialog dataset (Li et al., 2017), which is similar in structure to the BEA task’s dataset. Liu et al. (2021) present an approach to make language model generations less politically biased using Reinforcement Learning. Toledo et al. (2023) demonstrate the viability of a Reinforcement Learning approach in text-based games. Notably they achieve improvements over the previous state of the art in this zero-shot setting. The task of aiding students is comparable due to the large number of possible topics and unforeseen behavior of students when interacting with either a human teacher or a machine teacher. While it is not specifically considered in this task and underrepresented in current research, likely due to the current state of research in this area, there is the possible danger of models becoming outdated in the future, possibly very quickly, as the world around us changes. A solution for this is of course to re-train the models on new data to update them, but a strong performance in a zero-shot setting circumvents this problem altogether, and Reinforcement Learning approaches show viability in this area.

### 3 Data

The training data provided for the task by the organizers consists of 2747 samples of student-teacher dialogues from the Teacher Student Chatroom Corpus (Caines et al., 2020, 2022). There are always two speakers, a student and a teacher, and they take turns talking. Each of the samples contains one response. Each dialogue turn is prefixed with *teacher:* or *student:*, respectively. We use the full input dialogue as the input text, separating each speaker turn by newline. The reference text is the teacher response that follows the input dialogue. We used the t5-base model as well as the gpt2 model from HuggingFace and their respective tokenizers. Table 1 shows the lengths of the official training set released for the task.

To avoid potential issues or the need to cut off samples from the test set we have padded all the in-

put tokens to a length of 256 tokens for our model. We note that the task description states that each passage is at most 100 tokens long. The difference in maximum lengths likely comes from our chosen tokenizers, which uses a different tokenizing strategy than the approach that was used to calculate the expected maximum length of 100 tokens. For the training process we used a 80/10/10 split for training-validation-testing of the released training data.

## 4 Approach

Below, we present the methods we developed to generate teacher responses in real-world samples of teacher-student interactions.

### 4.1 Reinforcement Learning in NLP

Our submission to the task leaderboard is a sequence-to-sequence-based model. The task is structured in a way that is suited for these kinds of models: Given an input sequence of student-teacher dialogues, the output is another sequence, the response of the teacher. The comparatively small size of the data set and simplicity of the data set allows fast prototyping and experimentation. One research area where problems are also often small is that of Reinforcement Learning (Sutton and Barto, 2018). While combining Reinforcement Learning with human feedback is an active field (Knox and Stone, 2008; Arumugam et al., 2019; Li et al., 2019; Christiano et al., 2023), it has only recently started being used in the field of NLP (Ziegler et al., 2019; Ouyang et al., 2022b; Lambert et al., 2022). Most importantly, the RL4LMs framework (Ramamurthy et al., 2023) has enabled the easy adaptation of RL approaches for NLP tasks. The authors have applied their framework to similar tasks, notably the IMDB review continuation, using the dataset by Maas et al. (2011). They achieved good results on this task using GPT2. They further report good results using T5 (Raffel et al., 2020) for a summarization task on news (Hermann et al., 2015) as well as the CommonGen task (Lin et al., 2019).

### 4.2 T5

In the spirit of research we have initially decided to use T5 for this task instead of following the findings of the authors and using GPT2 due to the task’s similarity to the IMDB task. The compatibility of our chosen model with both being fine-tuned with

Reinforcement Learning as well as being usable in the RL4LMs framework has been demonstrated on a different task, so we conclude that our approach, while admittedly unusual, is not entirely unfounded in prior research.

### 4.3 GPT-2

Due to the relatively low ranking on the leaderboard of our T5 model we have additionally fine-tuned a GPT-2 checkpoint from the HuggingFace repository, with 124 million parameters, after the task concluded. As such this model was not submitted to the leaderboard. We include the configuration used for the training of both models in the appendix.

### 4.4 Algorithm

We follow the findings of Ramamurthy et al. (2023) and use their NLPO algorithm for the policy optimization during training. The performance of this algorithm is reported as the highest. It is an extension of the PPO algorithm (Schulman et al., 2017) and masks unlikely actions to reduce the action space. In the context of language generation this means masking next tokens whose cumulative probability is below a certain threshold. This reduction of the action space is important in the context of natural language problems as the action space in these contexts can be quite large. In the context of Reinforcement Learning a policy is a probability distribution over actions given a state. In our approach the policy is the language model being fine-tuned. The state is the generated tokens and the action is the next token to be generated in a language generation setting. Considering a language model itself to be a policy is a concept that has been used before in Liu et al. (2021) but is not widespread yet.

### 4.5 Reward Function

As our reward function we have chosen a pragmatic approach. We decided to use one of the metrics used in the evaluation as the reward function, as that should allow us to train the model to achieve a high score. The possibility of doing this showcases an advantage that a Reinforcement Learning-based approach has over other, more traditional approaches (both classic Machine Learning and Deep Learning) in the field of NLP: To lessen the gap between the evaluation criteria and the loss during training. Approaches for this problem exist (Song et al., 2016; Casas Manzanares et al., 2018)

but it remains an open problem. This mismatch can be avoided by using Reinforcement Learning, and, in theory, should allow a high performance on a variety of tasks. Ramamurthy et al. (2023) report that the quality of the reward function has a greater effect on the performance of the model than the amount of training data. To keep our reward function clear we have opted to use only one metric as the reward signal, as opposed to combining all the evaluation metrics into one function that calculates a scalar value. We experimented with using the average of all the evaluation metrics as the reward but empirically found quickly that this does not yield good performance and have not pursued this direction further. The metrics for the BEA task are BERTScore (Zhang et al., 2020) and DialogRPT updown, human vs. rand and human vs. machine scores (Gao et al., 2020). We wanted to avoid the potential issue of reward hacking and thus decided not to use the updown score as a metric, as it seemed potentially prone to that issue. The other two DialogRPT scores were eliminated due producing very high scores (above 0.95) even early on during training and thus are unlikely to be useful as reward signals, as any improvements that the model learns could only lead to marginal increases in reward. For this reason we have chosen to use the BERTScore, specifically the F1, as our reward function.

## 5 Results

In Table 2 we present the outputs by a zero-shot t5-base model, our fine-tuned t5-base model and our fine-tuned GPT-2 model. Model output were not trimmed or modified. We note that the both the fine-tuned T5 and GPT-2 include prefixes in their responses in some cases. The GPT-2 model is especially prone to outputting a "student:" response, which is not the goal of the task. This does not have an overly negative effect on the evaluation metrics however. Further investigation of the alignment of the task metrics with the stated goal of generative models assuming the role of teacher in student-teacher dialogues is recommended for this reason. Prompting the models by using the dialogue and adding a "teacher:" prompt at the end guided the models towards first writing a teacher response and only after that, on occasion, further student responses. To minimize assumptions and to modifying the task to improve our results we have not pursued the evaluation in this direction, and

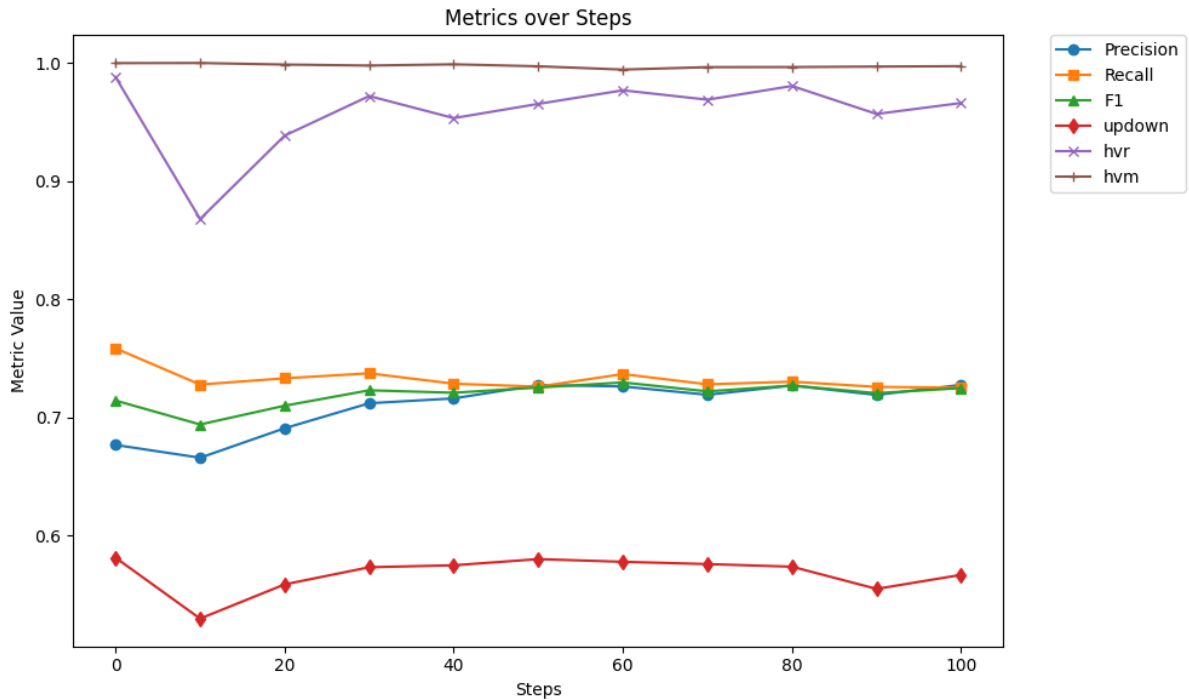


Figure 1: Metrics during the training process on the validation set for the GPT-2 model.

instead evaluated the models only on their output when given a dialogue, without any further prompting or modification.

### 5.1 Training Performance

Figure 1 shows the scores our GPT-2 model has achieved during the training process on the validation set. The scores of the trained model as well as zero-shot performance on the validation set are reported in Table 3. Due to an error the validation set splits were not pure during the training process of the T5 model and we do not include it in the graphic above.

### 5.2 Test Set Performance

We present the results of the evaluation on the test set in Table 4. Model outputs were generated on the test data dialogues, with the prefixes included, and were not pruned. Models often included wrong prefixes such as "student:" in their response. We did not remove these or filter the outputs for the first "teacher:" response. GPT-2 responses were set to have a minimum length of 12 and a maximum length of 100.

## 6 Conclusion

In this work we have shown our Reinforcement Learning-based approach on the BEA 2023 Shared

Task. We have used a relatively simple approach and trained two models, t5-base with 220 million parameters and gpt2 with 124 million parameters. Despite the overall performance of the models being mixed we have achieved good results in some areas. The GPT-2 model has achieved a good performance on the task and is showing clear gains in terms of evaluation metrics over a zero-shot approach on the same data. This suggests that Reinforcement Learning-based fine-tuning of language models is a valid approach. According to previous work in the area the model performance when fine-tuned with Reinforcement Learning is strongly influenced by the quality of the reward function. Our approach to this task was very basic and leaves room for improvement, which we believe can be achieved by using both higher quality models instead of relatively small ones with few parameters as well as an improved reward function that makes use of multiple evaluation metrics.

ID	Zero-Shot t5-base	t5-base with RL	GPT-2 with RL	Correct Labels
test_0001	student: 'extreme weather' and 'extreme climate'!	yes!	student: Yes, that's right.\n student: Solar/wind energy is a great idea	yes fantastic - so one answer is to 'reduce carbon emissions' (Collocation!) - what kind of energy is e.g. solar/wind
test_0002	Maybe some of them were too young!	To please the researchers?	teacher:\n teacher:\n teacher:\n	Could you make a sentence with a modal? Some of them ___ ___ ___ too young
test_0003	student: or - my train leaving time at 9:30.	ok?	student:...\n student:...\n student:...\n'	So: 'I'm going home tomorrow - my train ___ at 9:30' One word...what is it?
test_0004	so my time was faster/quicker/better than yesterday!	OK!	teacher: ok... so my time was faster.\n teacher:	OK so one more: My personal best for 5k is 15:03 (along time ago) .... So 15:03 is the ___ I ___ the 5k.??
test_0005	student: right?	teacher:...does it matter?	student: yes, that's right.\n student:...	OK see you in 15 mins ok?

Table 2: Comparison of responses from trained model, untrained model, and the correct labels, for our T5 model.

Model	BERTScore			DialogRPT (Avg/Max)		
	Precision	Recall	F1	updown	hvr	hvm
GPT-2 (zero-shot)	0.65	0.69	0.67	0.65/0.84	0.99/1.0	1.0/1.0
GPT-2 (RL)	0.73	0.72	0.72	0.57/0.80	0.97/1.0	0.90/1.0

Table 3: Evaluation metrics for the fine-tuned GPT-2 model and zero-shot performance of the untrained model on the validation set.

Model	BERTScore			DialogRPT (Avg/Max)		
	Precision	Recall	F1	updown	hvr	hvm
T5 (zero-shot)	0.71	0.69	0.70	0.62/0.85	0.98/1.0	0.95/1.0
T5 (RL, submitted)	0.76	0.65	0.70	0.50/0.70	0.92/1.0	0.88/1.0
GPT-2 (zero-shot)	0.68	0.65	0.66	0.67/0.85	1.0/1.0	0.99/1.0
GPT-2 (RL)	0.77	0.66	0.71	0.59/0.80	0.98/1.0	0.96/1.0

Table 4: Evaluation metrics on the official test set. Scores were calculated using the released labels. Model inputs included the speaker prefix. Outputs were not pruned or filtered and often included a prefix.

## References

- Dilip Arumugam, Jun Ki Lee, Sophie Saskin, and Michael L Littman. 2019. Deep reinforcement learning from policy-dependent human feedback. *arXiv preprint arXiv:1902.04257*.
- Andrew Caines, Helen Yannakoudakis, Helen Allen, Pascual Pérez-Paredes, Bill Byrne, and Paula Buttery. 2022. The teacher-student chatroom corpus version 2: more lessons, new annotation, automatic detection of sequence shifts. In *Swedish Language Technology Conference and NLP4CALL*, pages 23–35.
- Andrew Caines, Helen Yannakoudakis, Helena Edmondson, Helen Allen, Pascual Pérez-Paredes, Bill Byrne, and Paula Buttery. 2020. The teacher-student chatroom corpus. *arXiv preprint arXiv:2011.07109*.
- Noé Casas Manzanares, José Adrián Rodríguez Fonolosa, and Marta Ruiz Costa-Jussà. 2018. A differentiable bleu loss. analysis and first results. In *ICLR 2018 Workshop Track: 6th International Conference on Learning Representations: Vancouver Convention Center, Vancouver, BC, Canada: April 30-May 3, 2018*.
- Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martić, Shane Legg, and Dario Amodei. 2023. [Deep reinforcement learning from human preferences](#).
- Xiang Gao, Yizhe Zhang, Michel Galley, Chris Brockett, and Bill Dolan. 2020. [Dialogue response ranking training with large-scale human feedback data](#).
- Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.
- W Bradley Knox and Peter Stone. 2008. Tamer: Training an agent manually via evaluative reinforcement. In *2008 7th IEEE international conference on development and learning*, pages 292–297. IEEE.
- Nathan Lambert, Louis Castricato, Leandro von Werra, and Alex Havrilla. 2022. Illustrating reinforcement learning from human feedback (rlhf). *Hugging Face Blog*. <https://huggingface.co/blog/rlhf>.
- Guangliang Li, Randy Gomez, Keisuke Nakamura, and Bo He. 2019. Human-centered reinforcement learning: A survey. *IEEE Transactions on Human-Machine Systems*, 49(4):337–349.
- Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. [DailyDialog: A manually labelled multi-turn dialogue dataset](#). In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.
- Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2019. CommonGen: A constrained text generation challenge for generative commonsense reasoning. *arXiv preprint arXiv:1911.03705*.
- Ruibo Liu, Chenyan Jia, Jason Wei, Guangxuan Xu, Lili Wang, and Soroush Vosoughi. 2021. [Mitigating political bias in language models through reinforced calibration](#).
- Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022a. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.
- Siqi Ouyang, Rong Ye, and Lei Li. 2022b. [On the impact of noises in crowd-sourced data for speech translation](#). In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 92–97, Dublin, Ireland (in-person and online). Association for Computational Linguistics.
- 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.
- Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2023. [Is reinforcement learning \(not\) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization](#).
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. [Proximal policy optimization algorithms](#).
- Yang Song, Alexander G. Schwing, Richard S. Zemel, and Raquel Urtasun. 2016. [Training deep neural networks via direct loss minimization](#).
- Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.
- Anaïs Tack, Ekaterina Kochmar, Zheng Yuan, Serge Bibauw, and Chris Piech. 2023. The BEA 2023 Shared Task on Generating AI Teacher Responses in Educational Dialogues. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications*, page to appear, Toronto, Canada. Association for Computational Linguistics.

Edan Toledo, Jan Buys, and Jonathan Shock. 2023. [Policy-based reinforcement learning for generalisation in interactive text-based environments](#). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1230–1242, Dubrovnik, Croatia. Association for Computational Linguistics.

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

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

## A Appendix

We include our RL4LMs configurations used for training. The configuration seen in Figure 2 shows the configuration for the submitted T5 model. The reward function `bertscore_bea` is the F1 BERTScore, using the "distilbert-base-uncased" model, with the prefixes removed before the rewards are calculated. Figure 3 shows the configuration for the GPT-2 model. The reward function does not remove the prefixes before calculating the reward.

```

tokenizer:
  model_name: t5-base
  padding_side: left
  truncation_side: left
  pad_token_as_eos_token: False

reward_fn:
  id: bertscore_bea
  args:
    language: en

datapool:
  id: bea_full_seq2seq_splits_onlyResponse
  args:
    file_path: "/data/bea/data/release_1_train_dev/train_with-reference.jsonl"

env:
  n_envs: 1
  args:
    max_prompt_length: 256
    max_episode_length: 100
    terminate_on_eos: True
    prompt_truncation_side: "right"
    context_start_token: 0

alg:
  id: nlpo
  args:
    n_steps: 128
    batch_size: 64
    verbose: 1
    learning_rate: 0.00001
    n_epochs: 5
    ent_coef: 0.0
    gae_lambda: 0.9
    vf_coef: 0.1
  kl_div:
    coeff: 0.02
    target_kl: 2
  policy:
    id: maskable_seq2seq_lm_actor_critic_policy
    args:
      model_name: t5-base
      apply_model_parallel: True
      mask_type: "learned_top_p"
      top_mask: 0.9
      target_update_iterations: 20
      generation_kwargs:
        do_sample: True
        min_length: 20
        top_k: 200
        max_new_tokens: 100 # this must align with env's max steps

train_evaluation:
  eval_batch_size: 100
  n_iters: 100
  eval_every: 10
  save_every: 10
  metrics:
    - id: bertscore_bea
      args:
        language: en
    - id: bert_score
      args:
        language: en

```

Figure 2: RL4LMs configuration used for training the T5 model.

```

tokenizer:
  model_name: gpt2
  padding_side: left
  truncation_side: left
  pad_token_as_eos_token: True

reward_fn:
  id: bertscore_bea_distil
  args:
    language: en

datapool:
  id: bea_full_seq2seq_splits_onlyResponseNoShuffle
  args:
    file_path: "/data/bea/data/release_1_train_dev/train_with-reference.jsonl"

env:
  n_envs: 1
  args:
    max_prompt_length: 256
    max_episode_length: 100
    terminate_on_eos: True

alg:
  id: nlpo
  args:
    n_steps: 128
    batch_size: 64
    verbose: 1
    learning_rate: 0.00001
    n_epochs: 5

kl_div:
  coeff: 0.1
  target_kl: 1.0
policy:
  id: maskable_causal_lm_actor_critic_policy
  args:
    model_name: gpt2
    apply_model_parallel: True
    top_mask: 0.9
    min_tokens_to_keep: 100
    mask_type: 'learned_top_p'
    target_update_iterations: 5
    generation_kwargs:
      do_sample: True
      min_length: 12
      max_new_tokens: 100

train_evaluation:
  eval_batch_size: 100
  n_iters: 100
  eval_every: 10
  save_every: 10
  metrics:
    - id: bertscore_bea_distil
      args:
        language: en

```

Figure 3: RL4LMs configuration used for training the GPT-2 model.



# Assessing the efficacy of large language models in generating accurate teacher responses

Yann Hicke, Abhishek Masand, Wentao Guo, Tushaar Gangavarapu  
Cornell University

## Abstract

(Tack et al., 2023) organized the shared task hosted by the 18th Workshop on Innovative Use of NLP for Building Educational Applications on generation of teacher language in educational dialogues. Following the structure of the shared task, in this study, we attempt to assess the generative abilities of large language models in providing informative and helpful insights to students, thereby simulating the role of a knowledgeable teacher. To this end, we present an extensive evaluation of several benchmarking generative models, including GPT-4 (few-shot, in-context learning), fine-tuned GPT-2, and fine-tuned DialoGPT. Additionally, to optimize for pedagogical quality, we fine-tuned the Flan-T5 model using reinforcement learning. Our experimental findings on the Teacher-Student Chatroom Corpus subset indicate the efficacy of GPT-4 over other fine-tuned models, measured using BERTScore and DialogRPT.

We hypothesize that several dataset characteristics, including sampling, representativeness, and dialog completeness, pose significant challenges to fine-tuning, thus contributing to the poor generalizability of the fine-tuned models. Finally, we note the need for these generative models to be evaluated with a metric that relies not only on dialog coherence and matched language modeling distribution but also on the model’s ability to showcase pedagogical skills.

## 1 Introduction

The advent of powerful open-source generative language models such as GPT-2 (Radford et al., 2019), T5 (Raffel et al., 2020), OPT (Zhang et al., 2022), BLOOM (Scao et al., 2022), Flan-T5 (Chung et al., 2022) or LLAMA (Touvron et al., 2023) has led to significant developments in conversational agents, opening avenues for various applications in education (Wollny et al., 2021). Such AI-driven educational dialogues offer the potential for skill improvement and personalized learning

experiences, with intelligent tutoring systems increasingly gaining traction (Bibauw et al., 2022). However, deploying AI-based teachers in the educational ecosystem demands the careful modeling and evaluation of these agents to ensure their capability to address critical pedagogical concerns.

(Tack and Piech, 2022) created the AI teacher test challenge which follows the recommendations from (Bommasani et al., 2021) (pp. 67-72) stating that, if we want to put generative models into practice as AI teachers, it is imperative to determine whether they can (a) speak to students like a teacher, (b) understand students, and (c) help students improve their understanding.

Taking inspiration from the AI teacher test challenge which asks whether state-of-the-art generative models are good AI teachers, capable of replying to a student in an educational dialogue this paper seeks to investigate the applicability of reinforcement learning (RL) techniques in the generation of AI teacher responses within educational dialogues. The AI teacher test challenge emphasizes the need for a systematic evaluation of generative models to ensure that they can effectively communicate with students, comprehend their needs, and facilitate their academic improvement. Can we guide the language generator with RL to help it focus on these pedagogical requirements?

(Tack et al., 2023) organized the shared task hosted by the 18th Workshop on Innovative Use of NLP for Building Educational Applications on generation of teacher language in educational dialogues. Following the structure of the shared task, in this study, we aim to evaluate the potential of combining state-of-the-art generative language models with reinforcement learning algorithms to generate AI teacher responses in the context of real-world educational dialogues sourced from the Teacher Student Chatroom Corpus (Caines et al., 2020, 2022). The natural baselines for the task at hand are SOTA closed-source models such as GPT-4, and

fine-tuned open-source pre-trained models such as GPT-2 (Radford et al., 2019). We will evaluate these natural baselines before evaluating fine-tuned pre-trained models using RL techniques, that optimize for pedagogical quality.

By exploring the role of reinforcement learning in guiding the generation of AI teacher responses, we aim to advance the discourse on the utilization of conversational agents in educational settings and contribute innovative ideas to the ongoing shared task on the generation of teacher language in educational dialogues at the 18th Workshop on Innovative Use of NLP for Building Educational Applications.

The rest of this paper is structured as follows. Section 2 offers a comprehensive review of relevant literature in the areas of AI-driven educational dialogues and reinforcement learning-based language generation. Section 3 discusses the analysis and processing of the dataset prior to conducting any language modeling tasks. In Section 4, the proposed model and its methodology for generating AI teacher responses in educational interactions are introduced. Section 5 evaluates the effects of our approach on the quality and relevance of the generated AI teacher responses and highlights key observations. Finally, Section 6 concludes the paper and explores potential directions for future research.

## 2 Related Work

A variety of related literature exists in the realm of conversational teaching between a student and a teacher. In this section, we review several notable works addressing aspects of teacher-student dialogues, foundation models, and conversational datasets, which have contributed to the progress and understanding of generative models in educational contexts.

### Teacher-Student Dialogues

One prominent resource in educational dialogues is the National Council of Teachers of English (NCTE) dataset (Kane, 2015). It includes numerous examples of teacher-student interactions, which can serve as a valuable resource for the training and evaluation of generative models in an educational context.

The SimTeacher dataset (Cohen et al., 2020) is an assemblage of information obtained through a "mixed-reality" simulation platform. This unique environment aids beginner educators in

honing essential skills for classroom settings by employing student avatars managed by human actors. All aspiring teachers from a prominent public university participate in several brief simulation sessions throughout their educational preparation program, focusing on improving their ability to encourage more profound textual understanding among students. The original researchers annotated a variable called "quality of feedback" within the transcript, determining how effectively teachers proactively assist students.

In (Chen et al., 2019), we can find a dataset collected from an education technology company that provides on-demand text-based tutoring for math and science. With a mobile application, a student can take a picture of a problem or write it down and is then connected to a professional tutor who guides the student to solve the problem. The dataset represents, after some selection, 108 tutors and 1821 students. Each session is associated with two outcome measures: (1) student satisfaction scores (1-5 scale) and (2) a rating by the tutor manager based on an evaluation rubric (0-1 scale).

### Foundation Models

(Bommasani et al., 2021) provided a comprehensive analysis of the opportunities and risks of foundation models, including insights into their use in educational applications. They identified potential benefits, such as personalized learning and accessibility, while also highlighting the major risks, such as unfair biases and the generation of harmful content. This work establishes the need for carefully crafted benchmarks and evaluations to assess the potential of generative models in education.

The AI Teacher Test (Tack and Piech, 2022) builds on this idea by examining the performance of generative models such as GPT3 (Brown et al., 2020) and Blender (Roller et al., 2020) in generating appropriate and informative responses in a teacher-student dialogue.

Kasneci et al. (Kasneci et al., 2023) conducted an investigation to understand the effectiveness of ChatGPT (Team, 2022) as a tool for educational support. They analyzed the model's performance in a student-tutoring context, examining its ability to provide accurate, relevant, and engaging responses for learners. By identifying the strengths and weaknesses of ChatGPT in this specific setting, they contributed to a better understanding of how

generative models can be successfully deployed in educational applications. Our work builds on these foundations by evaluating the potential of combining reinforcement learning with generative models to enhance the performance of AI teacher agents in educational dialogues.

### Conversational Uptake

(Collins, 1982) introduced the concept of uptake as a way to comprehend the effectiveness of conversational responses in a teacher-student dialogue. It laid the groundwork for the evaluation of generative models in dialogues by taking into account the relevance and appropriateness of model-generated responses.

Demszky et al. (Demszky et al., 2021) further explored the concept of Conversational Uptake by proposing metrics to assess the success of responses in maintaining and advancing a conversation. By applying these metrics to AI-generated responses, their work contributes to the evaluation of models in realistic conversation settings, including teacher-student dialogues. Our work attempts to guide the language generation process with similar goals in mind. We hope to find proxies of pedagogical quality through NLP metrics such as BERTScore combined with DialogRPT.

We continue by reviewing the literature utilizing reinforcement learning as a guide for language generation.

### Reinforcement Learning for language generation

Policy gradient-based algorithms and their variants have been widely used in text generation to optimize sequence-level metrics (Ranzato et al., 2015; Shen et al., 2015; Norouzi et al., 2016; Pasunuru and Bansal, 2018). Off-policy Reinforcement Learning (RL) is also commonly used in dialogue applications where online interaction with users is expensive (Serban et al., 2017; Jaques et al., 2019). The main difference in our work is that we take advantage of demonstrations and design generic reward functions for generation tasks. We extend this concept to educational contexts by employing reinforcement learning to guide the generation of AI teacher responses in educational dialogues. We focus on optimizing the responses of fine-tuned generative models based on a reward system designed to enhance the pedagogical quality of the

generated responses. Recently, Ramamurthy et al. (Ramamurthy et al., 2022) explored the efficacy of using RL to optimize language models in several natural language processing tasks, including text classification, sentiment analysis, and language generation. They developed a library, RL4LMs, which provides a generic framework for deploying RL-based language models for various tasks. We build on top of the RL4LMs framework by adding a new task to its existing array of tasks which we hope can be added as a standard for any future RLHF benchmark.

## 3 Data

The shared task for BEA 2023 is based on the Teacher-Student Chatroom Corpus (TSCC) (Caines et al., 2020). This corpus comprises data collected from 102 chatrooms where English as a Second Language (ESL) teachers interact with students to work on language exercises and assess the students' English language proficiency.

### 3.1 Data Extraction and Format

From each dialogue in the TSCC, several shorter passages were extracted. Each passage is at most 100 tokens long, consisting of several sequential teacher-student turns (i.e., the preceding dialogue context) and ending with a teacher utterance (i.e., the reference response). These short passages are the data samples used in this shared task.

The data samples are formatted using a JSON structure inspired by the ConvoKit (Chang et al., 2020). Each training sample is represented as a JSON object with three fields:

- **id**: a unique identifier for the sample.
- **utterances**: a list of utterances corresponding to the preceding dialogue context. Each utterance is a JSON object with a "text" field containing the utterance and a "speaker" field containing a unique label for the speaker.
- **response**: a reference response, which corresponds to the final teacher's utterance. This utterance is a JSON object with a "text" field containing the utterance and a "speaker" field containing a unique label for the speaker.

Each test sample is represented as a JSON object that uses the same format as the training sample but excludes the reference response. As a result, each test sample has two fields:

- **id**: a unique identifier for the sample.
- **utterances**: a list of utterances, which corresponds to the preceding dialogue context. Each utterance is a JSON object with a "text" field containing the utterance and a "speaker" field containing a unique label for the speaker.

### 3.2 Data Distribution and Characteristics

The TSCC corpus is divided into three sets: train, dev, and test, each comprising 2747, 305 and 273 of the samples, respectively. The corpus has 3325 samples, and each sample has an average length of 7.52 turns, with about 7.33 tokens per turn on average. Table 1 presents a summary of the statistics of the TSCC corpus across the training, development, and testing sets.

The TSCC corpus exhibits several characteristics that are specific to educational dialogues and pose challenges to natural language generation models. For instance, the dialogues often include technical vocabulary and idiomatic expressions related to English language learning. Additionally, the dialogues can be highly varied in terms of topic, complexity, and participant proficiency. Finally, the dialogues can include challenging responses which are based on out-of-context information, posing challenges for conversational agents. These characteristics must be taken into consideration when selecting and evaluating generative models for the TSCC corpus.

### 3.3 Data Overlap and Challenges

It is worth noting that the released development and training sets in the TSCC dataset have some overlaps, as individual conversation samples within these sets have been generated by creating chunks from larger conversations. This overlap may lead to potential biases and overfitting when training and evaluating models on this dataset. However, the test set for the BEA 2023 shared task is free of overlaps, allowing for a more accurate assessment of the model’s performance in generating AI teacher responses.

The presence of overlaps in the development and training sets posed a challenge, as models inadvertently learned to predict teacher responses based on the similarities between the samples rather than genuinely understanding the context and dynamics of the teacher-student interaction. It is essential to be aware of this issue and devise strategies to mitigate the risks associated with such overlaps and

ensure that the models are robust and capable of handling diverse and unseen scenarios.

To ensure the validity of our model on the validation set, we employed an iterative inclusion process to create a train-val split without any overlap between them. This process involved carefully selecting and excluding samples from the training set that had any similarity or overlap with the samples in the development set. This approach aimed to minimize the risk of data leakage and ensure that our model was evaluated on a truly unseen set of dialogues.

## 4 Methods

The primary objective of our study is to investigate the potential of using in-context learning, supervised fine-tuning, and reinforcement learning to generate AI teacher responses in educational dialogues. Our proposed methods will be evaluated using the Teacher Student Chatroom Corpus (TSCC) dataset. In this section, we provide an overview of the three main parts of our methodology: in-context learning using GPT-4, supervised fine-tuning with existing models such as GPT-2 and DialoGPT, and supervised fine-tuning with Reinforcement Learning using the RL4LMs library (Ramamurthy et al., 2022).

### 4.1 In-context Learning

#### 4.1.1 GPT-4

As a preliminary step, we investigate the potential of in-context learning using GPT-4, a state-of-the-art language model. It generates educational dialogues based on its pre-trained knowledge, which has been acquired from a vast corpus of text data during its training process (the pre-training data might have included the test set; we will address this issue in the discussion section).

To evaluate the performance of GPT-4, we prompted GPT-4 in a few-shot fashion. We retrieved 5 most similar teacher-student conversations from the TSCC dataset and provided them to the model in addition to the current conversation and instructions about the model’s role as a teacher. Details about the prompt construction that helps guide the model toward generating suitable responses as a teacher can be found in the Appendix A.

Table 1: Summary of the statistics for the TSCC corpus across the train, dev, and test sets.

Dataset	Train	Dev	Test
Num Samples	2747	305	273
Avg Turns	7.7	7.92	5.23
Avg Tokens Per Turn	7.29	7.21	8.27

## 4.2 Supervised Fine-tuning

To further adapt pre-trained language models to the specific educational context and generate more accurate and context-aware teacher responses, we explore supervised fine-tuning using GPT-2 and DialoGPT models.

### 4.2.1 GPT-2

GPT-2 (Radford et al., 2019) is a decoder-only large language model pre-trained on WebText, and we used GPT-2 Large, which has 24 transformer decoder blocks with 774 million parameters.

We fine-tune the GPT-2 model (Radford et al., 2019) using the Huggingface Library on the Teacher Student Chatroom Corpus (TSCC) dataset. For each educational dialogue, we concatenated all dialogue turns into a single string with additional information of speaker roles i.e. students or teachers. As a result, the input to the GPT-2 model consists of a sequence of text representing the conversation history, culminating in the teacher’s response. We then finetuned GPT-2 Large (Radford et al., 2019) with a casual language modeling task. Details of the exact hyperparameters used during the fine-tuning process can be found in the Appendix.

After the fine-tuning process, we evaluated the fine-tuned GPT-2 model’s performance on the test set by comparing its generated teacher responses to reference responses, assessing the model’s ability to generate context-aware and educationally relevant responses in line with the teacher’s role in the TSCC dataset.

### 4.2.2 DialoGPT

DialoGPT (Zhang et al., 2019) is a dialogue model based on the GPT-2 architecture, specifically designed for generating conversational responses. DialoGPT is trained with 147M conversation pieces extracted from Reddit (Zhang et al., 2019), and it is trained with casual language modeling objectives with multi-turn dialogue. We adapt our training dataset with the same format as that of DialoGPT during pretraining and then prompt the DialoGPT to generate an educational dialogue of teachers in the validation set. After training, we follow the

same methodology for evaluation as GPT-2 which we discussed in the earlier section.

## 4.3 Supervised Fine-tuning with Reinforcement Learning

### 4.3.1 Flan-T5 Fine-tuned with RL4LMs

To optimize the generative models for pedagogical quality, we explore the use of reinforcement learning techniques in the fine-tuning process. We employ the RL4LMs library (Ramamurthy et al., 2022), which provides an efficient and scalable framework for reinforcement learning-based language model fine-tuning.

The RL4LMs library incorporates Proximal Policy Optimization (PPO) (Schulman et al., 2017) as the reinforcement learning algorithm, which is known for its stability and sample efficiency. The library also supports the integration of custom reward functions, allowing us to design rewards that encourage the generation of pedagogically sound teacher responses.

To implement the reinforcement learning-based fine-tuning, we first fine-tune the Flan-T5 (Chung et al., 2022) model on the TSCC dataset using supervised learning, as described in the previous section. Next, we utilize the RL4LMs library to fine-tune the model further using the PPO algorithm. We use an equal division of the F1 as calculated by the roberta-large version of BERTScore and DialogRPT-updown as the reward function. More Details about the reinforcement learning fine-tuning process can be found in the Appendix.

The subsequent evaluation of the fine-tuned Flan-T5 model reveals the benefits of incorporating reinforcement learning into the fine-tuning process, contributing to more context-aware, relevant, and pedagogically effective AI teacher responses.

## 5 Results

In this section, we present the results and discuss the performance of GPT-4, fine-tuned GPT-2, and fine-tuned DialoGPT models on the TSCC dataset. We analyze the strengths and weaknesses of each approach and provide insights into their potential

applications and limitations in an educational context.

### 5.1 GPT-4

The GPT-4 model, without fine-tuning on the TSCC dataset, demonstrates a relatively strong performance in generating educational dialogues. The generated teacher responses are generally fluent and contextually relevant, indicating that GPT-4 has a good understanding of the educational context based on its pre-trained knowledge. However, some limitations are observed in the model’s ability to generate accurate and pedagogically sound responses consistently.

The carefully crafted prompt provided to the model plays a crucial role in guiding GPT-4 toward generating suitable responses as a teacher. Although the model is capable of generating contextually relevant and linguistically correct responses, it may not always produce the most pedagogically sound or helpful responses for the students. This limitation highlights the importance of fine-tuning the model on a specific educational dataset, such as TSCC, to further enhance its performance in generating AI teacher responses.

Additionally, due to the nature of the dataset, where conversations were often cut off, the model sometimes lacked the full context needed to generate meaningful responses that accurately represented the ground truth. Despite this limitation, GPT-4’s responses were generally sensible and appropriate given the available context.

### 5.2 Finetuned GPT-2

We observe that compared with DialoGPT, GPT-2 usually generates longer and more formal responses, even with the same generation hyperparameters.

### 5.3 Finetuned DialoGPT

We observe that DialoGPT usually generates shorter and more vernacular responses. It fits better in a conversational setting, but sometimes the educational uptakes are not satisfactory since the responses are not guiding students to learn the language.

### 5.4 Finetuned Flan-T5 w/ RL

We observe that the results of Flan-T5 w/ RL on the validation set are really good suggesting that the model was able to hack the metrics designed as the reward. On the contrary, it is performing poorly on

the test set suggesting that it overfits the validation set. We hypothesize two reasons for this to be the case: the way conversations are split into chunks in the training dataset or the difference in distribution between the training set and the test set.

## 6 Discussion

Conversational agents have the potential to revolutionize the teaching landscape by addressing several challenges and enhancing the overall learning experience for both students and educators (Wollny et al., 2021). However, developing conversational agents that can behave like human teachers requires addressing several challenges (Tack and Piech, 2022).

**Data challenges.** As noted in the subsections above, the generations from the GPT-4 model outperformed all the fine-tuned models, with and without reinforcement learning. To this end, we put forward the proposition that an array of dataset features plays a crucial role in posing significant challenges to the fine-tuning process of generative models. These features include several dataset characteristics, including sampling, representativeness, prompt and response lengths, and dialogue completeness—upon manual inspection, we identified several dialogues to be cut off—pose serious challenges in achieving superior performance with fine-tuning. Furthermore, upon random inspection of the generations from the fine-tuned models, we identified that these models seem to have learned simple, generic, often inappropriate yet correct responses such as “thank you” and “okay.” While more recent language models have been shown to have high few-shot performance, we believe that fine-tuned models could adapt better to provide domain-specific responses in comparison. To achieve this, we emphasize the need for extending the current dataset to include longer prompts with more context.

It is important to acknowledge that these models might not be as effective as desired in their response generation due to these intricacies. The current efforts made by the research community to collect and build quality datasets encompassing enough information about the educational task to enable AI teacher generative models to fully generalize in any context is what we assess to be the main focus that the community should adopt

Table 2: Validation set results

Model	BERTScore	DialogRPT
GPT-4	0.82	0.69
Finetuned GPT-2 Large	<b>0.94</b>	0.63
Finetuned DialoGPT Large	<b>0.94</b>	0.64
Finetuned Flan-T5 w/ RL	0.89	<b>0.71</b>

Table 3: Test set results

Model	BERTScore	DialogRPT
GPT-4	<b>0.8</b>	<b>0.70</b>
Finetuned Flan-T5 w/ RL	0.66	0.34

[student]	someone plugged the charger in
[teacher]	that’s bad, charger must be ___?
[student]	umm . . .
[model]	(a) <b>plugged in</b> ← score: 0.91 (b) <b>disconnected</b> ← score: 0.90
[reference]	plugged out

Figure 1: An example dialog demonstrating that two opposing responses, (a) and (b), ranked alike using the BERTScore metric.

(Jarratt, 2023).

**Evaluation metrics.** In addition, we emphasize that to truly gauge the efficiency of these AI-powered teaching models, it is vital to go a step further and examine their ability to comprehend the unique nuances in the students’ queries and cater to their particular educational requirements. This implies the need for a pedagogically meaningful evaluation metric. We believe that it is crucial for the research community to embrace this as the second primary focus. While common evaluation metrics such as BERTScore and DialogRPT are commonly used in several language and dialog modeling tasks, it is important to note that these metrics were not fundamentally designed to capture the level of pedagogical meaningfulness in the generated responses. As an example, consider the dialog shown in Figure 1—depending on the given context, only one of the responses (option (b): disconnected) is correct, while both the responses

are ranked as equally correct by the BERTScore metric. Commonly-used domain-agnostic metrics often serve as a proxy for how coherent and human-like the generated responses are. However, for more goal-oriented tasks such as modeling teacher-student conversational dialogues, these metrics seem to fall short. This generalization gap becomes more apparent on analyzing the results from the fine-tuned Flan-T5 model with a feedback loop based on BERTScore and DialogRPT scores—despite the model performing significantly well on training and validation sets, it failed to generalize on unseen test data. In an effort to advance research on this front, we note the need for auxiliary training-level metrics, including the faithfulness of the generation to the true response, to ensure that the generations are context-aware and factually accurate (e.g., correct option (b) vs. incorrect option (a) in Figure 1).

**GPT-4 unknown pre-training data.** We understand that the use of GPT-4 as a baseline in our study presents challenges due to its unknown training data. Yet, whether GPT-4 has seen parts of the TSCC dataset during its pre-training or not, the improvement of performance compared to the reference with regard to the DialogRPT scores and human evaluation scores attached to the leaderboard of the shared task suggests that the potential of using such high-performing models in this domain warrants further exploration.

## 7 Conclusion

In this paper, we explored the potential of using large pre-trained language models and reinforcement learning for generating AI teacher responses in an educational context. We first presented a few-shot approach using the GPT-4 model, which demonstrated promising results in generating contextually relevant and fluent responses, but with limitations in generating pedagogically sound responses consistently. We then fine-tuned GPT-2 and DialoGPT on the TSCC dataset and evaluated their performance using BERTScore and DialogRPT metrics. We also proposed an approach using RL to optimize directly for pedagogical values. We hypothesized that several dataset characteristics (e.g., dialog completeness, sampling) pose challenges to achieving superior performance with fine-tuning. To this end, we recommend the extension of the dataset to include longer prompts with extended context. Finally, we also draw attention to the need for more domain-specific metrics (in both evaluation and reward-based training) in enabling the generation of accurate, context-aware, and factually correct teacher responses.

## References

- Serge Bibauw, Thomas François, and Piet Desmet. 2022. Dialogue systems for language learning: Chatbots and beyond. In *The Routledge handbook of second language acquisition and technology*, pages 121–135. Routledge.
- Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Andrew Caines, Helen Yannakoudakis, Helen Allen, Pascual Pérez-Paredes, Bill Byrne, and Paula Buttery. 2022. The teacher-student chatroom corpus version 2: more lessons, new annotation, automatic detection of sequence shifts. In *Proceedings of the 11th Workshop on NLP for Computer Assisted Language Learning*, pages 23–35.
- Andrew Caines, Helen Yannakoudakis, Helena Edmondson, Helen Allen, Pascual Pérez-Paredes, Bill Byrne, and Paula Buttery. 2020. The teacher-student chatroom corpus. *arXiv preprint arXiv:2011.07109*.
- Jonathan P Chang, Caleb Chiam, Liye Fu, Andrew Z Wang, Justine Zhang, and Cristian Danescu-Niculescu-Mizil. 2020. Convokit: A toolkit for the analysis of conversations. *arXiv preprint arXiv:2005.04246*.
- Guanliang Chen, Rafael Ferreira, David Lang, and Dragan Gasevic. 2019. Predictors of student satisfaction: A large-scale study of human-human online tutorial dialogues. *International Educational Data Mining Society*.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.
- Julie Cohen, Vivian Wong, Anandita Krishnamachari, and Rebekah Berlin. 2020. Teacher coaching in a simulated environment. *Educational evaluation and policy analysis*, 42(2):208–231.
- James Collins. 1982. Discourse style, classroom interaction and differential treatment. *Journal of reading behavior*, 14(4):429–437.
- Dorottya Demszky, Jing Liu, Zid Mancenido, Julie Cohen, Heather Hill, Dan Jurafsky, and Tatsunori Hashimoto. 2021. Measuring conversational uptake: A case study on student-teacher interactions. *arXiv preprint arXiv:2106.03873*.
- Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. 2019. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*.
- Daniel Jarratt. 2023. *Chatgpt: The double-edged sword of ai in education*.
- Thomas Kane. 2015. *National Center for Teacher Effectiveness Main Study*.
- Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günemann, Eyke Hüllermeier, et al. 2023. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103:102274.
- Mohammad Norouzi, Samy Bengio, Navdeep Jaitly, Mike Schuster, Yonghui Wu, Dale Schuurmans, et al. 2016. Reward augmented maximum likelihood for neural structured prediction. *Advances In Neural Information Processing Systems*, 29.
- Ramakanth Pasunuru and Mohit Bansal. 2018. Multi-reward reinforced summarization with saliency and entailment. *arXiv preprint arXiv:1804.06451*.



- 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. *The Journal of Machine Learning Research*, 21(1):5485–5551.
- Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2022. Is reinforcement learning (not) for natural language processing?: Benchmarks, baselines, and building blocks for natural language policy optimization. *arXiv preprint arXiv:2210.01241*.
- Marc’Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.
- Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M Smith, et al. 2020. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*.
- Tevan Le Scao, Angela Fan, Christopher Akiki, Elie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- 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*.
- Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2015. Minimum risk training for neural machine translation. *arXiv preprint arXiv:1512.02433*.
- Anais Tack, Ekaterina Kochmar, Zheng Yuan, Serge Bibauw, and Chris Piech. 2023. The BEA 2023 Shared Task on Generating AI Teacher Responses in Educational Dialogues. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications*, page to appear, Toronto, Canada. Association for Computational Linguistics.
- Anais Tack and Chris Piech. 2022. The ai teacher test: Measuring the pedagogical ability of blender and gpt-3 in educational dialogues. *arXiv preprint arXiv:2205.07540*.
- OpenAI Team. 2022. Chatgpt: Optimizing language models for dialogue.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Sebastian Wollny, Jan Schneider, Daniele Di Mitri, Joshua Weidlich, Marc Rittberger, and Hendrik Drachler. 2021. Are we there yet?-a systematic literature review on chatbots in education. *Frontiers in artificial intelligence*, 4:654924.
- Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.
- Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

## A Appendix

### A.1 GPT-4 Prompt Construction

To evaluate the performance of GPT-4, we provided it with a few-shot prompt that includes a selection of similar teacher-student conversations from the TSCC dataset. This approach helps guide the model toward generating suitable responses as a teacher. The prompt is constructed as follows:

- We direct the system role to act as a teacher and encourage learning by using the prompt as given below.
- Retrieve the 5 most similar teacher-student conversations from the TSCC dataset. This is done by computing the cosine similarity between the input conversation context and the current conversation context in the dataset using embeddings generated by the text-embedding-ada-002 model.
- Concatenate the selected conversations with the input conversation, separated by special tokens to indicate the beginning and end of a new sample conversation.

This prompt construction aims to provide GPT-4 with the necessary context and guidance to generate accurate and pedagogically relevant responses in the context of teacher-student dialogues. The prompt is designed as follows:

You are acting as a teacher, and you are helping a student learn. Be patient, helpful, and kind. Don't be superimposing; give short responses to encourage learning. Make the student feel comfortable and confident, and help them learn. Now, join the following conversation: <conversation context>

The prompt is designed using the following directives in mind:-

- We instruct the system with several indicators to act as a teacher and provide helpful advice to the student.
- To mitigate the challenge of generating teacher-like responses, we advise the model to be patient, kind, and helpful to the student.
- Through the directive to keep responses short and encouraging, we guide the model toward

generating suitable responses that might help the student learn effectively.

- The model is also instructed to make the student feel comfortable and confident in their learning process, providing an overall supportive environment for the student.
- Finally, the conversation context is provided to the model to set the context for the given student query, allowing the model to generate appropriate responses given the conversation context.

Combining all these aspects together, we aim to guide the model toward generating contextually relevant and pedagogically meaningful responses in the given teacher-student dialogue.

We use the following hyperparameters for querying the GPT-4 model:

- Model: gpt-4-0314
- Temperature: 1
- Max Tokens: 100
- Top p: 1

### A.2 Fine-tuning Exact Parameters

For our supervised fine-tuning experiments, we used the following hyperparameters:

#### A.2.1 GPT-2

- Learning rate: 1e-5
- Batch size: 32
- Epochs: 10
- Max sequence length: 1024
- Optimizer: AdamW
- Scheduler: linear learning rate scheduler

#### A.2.2 DialoGPT

- Learning rate: 1e-5
- Batch size: 32
- Epochs: 10
- Max sequence length: 1024
- Optimizer: AdamW
- Scheduler: linear learning rate scheduler

### A.3 Supervised Fine-tuning with Reinforcement Learning Details

To implement the reinforcement learning-based fine-tuning using the RL4LMs library, we first fine-tuned the Flan-T5 model on the TSCC dataset using supervised learning. After this initial fine-tuning step, we utilized the RL4LMs library to fine-tune the model further using reinforcement learning. We used an equal division of the BERTScore and DialogRPT as the reward function to optimize the model for pedagogical quality. The following hyperparameters were used for the reinforcement learning fine-tuning process:

- Learning rate: 1e-6
- Batch size: 64
- Epochs: 5
- Max prompt length: 512
- Max episode length: 100
- Optimizer: AdamW
- Scheduler: linear learning rate scheduler

The YAML file for the RL4LMs script is as follows:

```
tokenizer:
del_name: google/flan-t5-small
dding_side: left
uncation_side: left
d_token_as_eos_token: False
rd_fn:
: dialog_rpt_bert
gs:
BERTScore_coeff: 0.5
DialogRPT_coeff: 0.5
pool:
: bea
uncate: False
gs: {}

envs: 1
gs:
max_prompt_length: 100
max_episode_length: 20
terminate_on_eos: True
context_start_token: 0
prompt_truncation_side: "right"
```

```
: ppo_separate
gs:
n_steps: 20
batch_size: 64
verbose: 1
learning_rate: 0.000001
clip_range: 0.2
n_epochs: 1
value_update_epochs: 3
# batchify: False
gae_lambda: 0.95
gamma: 0.99
ent_coef: 0.01
_div:
coeff: 0.001
target_kl: 2.0
licy:
id: seq2seq_lm_actor_critic_policy
args:
model_name: google/flan-t5-small
apply_model_parallel: True
prompt_truncation_side: "right"
generation_kwargs:
do_sample: True
top_k: 0
min_length: 9
max_new_tokens: 20
n_evaluation:
al_batch_size: 64
iters: 200
al_every: 20
ve_every: 10
trics:
- id: bert_score
args:
language: en
- id: dialog_rpt
args:
model_name: "microsoft/DialogRPT
-updown"
label_ix: 0
batch_size: 1
# - id: uptake
# args:
# model_name: None
# label_ix: 0
# batch_size: 1
neration_kwargs:
num_beams: 5
min_length: 9
max_new_tokens: 20
```

# RETUYT-InCo at BEA 2023 Shared Task: Tuning Open-Source LLMs for Generating Teacher Responses

Alexis Baladón and Ignacio Sastre and Luis Chiruzzo and Aiala Rosá

Facultad de Ingeniería  
Universidad de la República  
Montevideo, Uruguay

{alexis.baladon, isastre, luischir, aialar}@fing.edu.uy

## Abstract

This paper presents the results of our participation in the BEA 2023 shared task, which focuses on generating AI teacher responses in educational dialogues. We conducted experiments using several Open-Source Large Language Models (LLMs) and explored fine-tuning techniques along with prompting strategies, including Few-Shot and Chain-of-Thought approaches. Our best model was ranked 4.5 in the competition with a BertScore F1 of 0.71 and a DialogRPT final (avg) of 0.35. Nevertheless, our internal results did not exactly correlate with those obtained in the competition, which showed the difficulty in evaluating this task. Other challenges we faced were data leakage on the train set and the irregular format of the conversations.

## 1 Introduction

Nowadays, with the important development of Large Language Models (LLM) and their great generative power, the interest in the development of chatbots that simulate interactions between humans has increased. In particular, in the educational domain, the use of chatbots seems to have interesting benefits, such as their potential for adaptive learning, tailored to each student, or their permanent availability (Bibauw et al., 2022).

The contributions of these tools to learning are not yet clear (Wollny et al., 2021). In their review of the area, these authors conclude that the development of chatbots is usually based on technological criteria, but the focus has not yet been placed on their pedagogical contributions in terms of learning improvements.

However, there is some evidence that for language learning in particular, these tools bring certain benefits (Bibauw et al., 2022), mainly for students at initial levels. It should be noted that in the case of language teaching, interaction with the agent is in itself an instance of learning practice.

One aspect to be studied in the development of educational chatbots is their ability to understand students needs and respond with the style that teachers, trained to educate, use to address their students (Bommasani et al., 2021). Although current LLMs show great capacity for language generation and for providing relevant -although not always correct or true- answers to different types of queries, it is important to study whether these models can be used in an educational context, being able to respond to a student by simulating a dialogue with a teacher. (Tack and Piech, 2022) propose such an evaluation called the AI teacher test challenge.

This paper presents the RETUYT-InCo submission to the BEA 2023 shared task (Tack et al., 2023) on generating teacher responses in educational dialogues. In this work, we analyze some particularities of the dataset used in the competition, we describe the approaches we made to solving the problem, and we present the results we obtained, together with an analysis and discussion of future steps.

## 2 Data analysis

The following study aims to understand the patterns and characteristics of the conversations between teachers and students, which will be crucial for training a chatbot to generate appropriate responses.

### 2.1 Dataset content

First, it is important to consider the description provided on the official BEA Shared Task webpage<sup>1</sup> and the source of the corpus used in this study. According to the information available, the corpus consists of extracts from 102 different chatrooms where an English teacher engages in language exercises and assesses the English language proficiency of the students (Caines et al., 2020). Each

<sup>1</sup><https://sig-edu.org/sharedtask/2023>

extract comprises a series of **utterances**, representing turns by the teacher and the student, along with a **response** that, as per the competition prompt, always originates from the teacher. This distinction is vital as the objective is not simply to continue a conversation but to respond from the perspective of a teacher.

Secondly, upon inspecting the corpus, it was revealed that the dataset contained additional sets of conversations beyond the original composition, as described in the corpus paper (Caines et al., 2020) and the provided website, which stated a total of 102 conversations. Hence, we assumed the corpus was composed with a set of extracts from each of those conversations, implying the data inside the corpus is not completely dependent. Interestingly, during the examination of the training corpus, numerous tuples were found to be partially duplicated, indicating that the conversations in the training set were derived from overlapping segments of the same original conversations. This issue is critical due to two main reasons. First, it is important to note that each teacher’s response does not correspond to the final utterance of the entire conversation but rather the last utterance within an extract from the conversation (similarly for the first utterance). Moreover, this poses a significant challenge when it comes to the typical validation approach of partitioning the dataset, as it is not immediately evident how to separate each conversation in a manner that prevents data leakage across corpus partitions without hindering the model’s training.

## 2.2 Other relevant findings

There are several noteworthy characteristics of the dataset to consider. Firstly, one of the initial examples showcased on the official website features a student attempting to solve a task involving filling a gap with a word or short phrase (see Fig. 1). However, upon inspecting the number of conversations that contain at least one underscore character (`_`), it was found that only 14.89% of them met this criterion. Consequently, while this restriction does not significantly impact the further architecture of the model, it is worth mentioning that incorporating this aspect could potentially enhance the model’s performance in future work.

Furthermore, some tasks within the dataset involve choosing between two options (a) or (b) type questions. However, due to the fact that these types of questions account for less than 1% of the total

corpus, the decision was made not to thoroughly analyze them in this study.

```
[ DIALOGUE CONTEXT ]
Teacher: Yes, good! And to charge it up, you need to _ it _
Student: _
Teacher: connect to the source of electricity
Student: i understand
Teacher: plug it _?
Student: in

[ REFERENCE RESPONSE ]
Teacher: yes, good. And when the battery is full, you need to _ (disconnect it)
```

Figure 1: Example of conversation extract

In addition, an examination of the dataset’s tags reveals a variety of categories, including `<STUDENT>`, `<TEACHER>`, `<ANOTHER STUDENT>`, `<CAT’S NAME>`, `<LIZARD’S NAME>`, and others. Notably, students and teachers represent over 90% of the tags. The presence of specific names and references to animals suggests that the dataset covers a wide range of topics related to conversations between teachers and students. A table displaying the most frequent tags count can be found in Table 1.

Tag	Count
<code>&lt;STUDENT&gt;</code>	868
<code>&lt;TEACHER&gt;</code>	141
<code>&lt;ANOTHER STUDENT&gt;</code>	19
<code>&lt;CAT’S NAME&gt;</code>	18
<code>&lt;LIZARD’S NAME&gt;</code>	17
<code>&lt;STUDENT’S SHORT NAME&gt;</code>	7
<code>&lt;CAT’S NAME1&gt;</code>	5
<code>&lt;STUDENT’S FULL NAME&gt;</code>	5
<code>&lt;LIZARD’S NAME’S&gt;</code>	4
<code>&lt;TEACHER’S NAME&gt;</code>	3

TABLE 1: 10 Most Frequent Tags in the Dataset

## 2.3 Proportion of conversation utterances

In addition to examining other aspects of the corpus, it is important to analyze whether the conversations exhibit any form of imbalance. Intuitively, one might expect the student to be more hesitant in their participation due to a lack of confidence, or conversely, the teacher may encourage the student to contribute more in order to facilitate learning. Therefore, the rate of text length expressed by each participant was assessed using two different measures: the length of tokens and the number of conversation turns.

To tokenize the sentences in the dataset, we used NLTK’s wordtokenize function (Bird et al., 2009). To understand the distribution of tokens (see Fig. 2, the analysis considered the token count for each part of the conversation, namely the teacher, the student, and both. The teacher’s responses had an average of 11.18 tokens, with a standard deviation of 9.37. The student’s responses had an average of 6.00 tokens, with a standard deviation of 6.49. When considering both parts of the conversation, the average token count was found to be 9.07. These findings suggest that the model should generate responses that are generally longer than those found in the dataset.

Subsequently, it was measured the same proportion taking only into consideration the number of utterances by each speaker. The analysis indicates that teachers account for 59.47% of the total conversation turns. However, it is important to acknowledge that this imbalance in the data is a direct consequence of the last tuple always being the teacher’s response. It is also worth highlighting that the turns do not always follow an alternating pattern based on the speaker, as there are instances where the same speaker appears consecutively. This deviation from the typical conversational pattern can present a challenge when training conversational chatbots that rely on alternating inputs from different speakers.

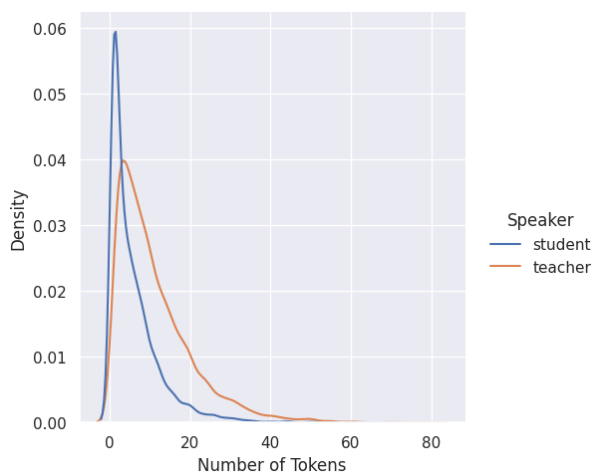


Figure 2: Student and Teacher’s token distribution

### 3 Experiments

This section described the systems implemented to solve the task.

```
### INSTRUCTION: You are an English teacher helping a student on their learning process. Given a conversation, generate the next teacher response.

### CONVERSATION:
teacher: could you finish the sentence, please?
student: I'll stay at home
teacher: great!
teacher: what about
teacher: If I ____ in the USA now, I ____ eat cheeseburgers for breakfast!
teacher: Silly example, I know, just for the grammar
student: Have been/ will
teacher: How about 'If I were in the USA now, I would eat...'

### RESPONSE:
teacher:
```

Figure 3: Prompt used with Alpaca LoRA applied to the example with id *train\_1504* from the training set.

#### 3.1 Using pretrained Large Language Models

Our first approach was trying out open source pretrained Large Language Models (LLMs), such as the model LLaMA (Touvron et al., 2023) and a fine-tuned version for following instructions available in Hugging Face, Alpaca LoRA<sup>2</sup>.

The dataset used for the fine-tuning of Alpaca LoRA is the one provided in (Taori et al., 2023), where each example is composed of three sections (the second is optional): *Instruction*, where the task is described, *Input*, which is an optional context for the task and *Response*, which is the answer to the instruction.

We designed a prompt following this format but we adapted it to integrate the whole conversation to the context. An specific instruction was designed for this task, and it is provided in the *Instruction* section. The input section was changed for a conversation section, where the utterances are presented in a classical chat format. The response section always starts with “teacher:”, influencing the model to generate a continuation for the conversation as a teacher. An example is presented in Fig. 3.

Following this experiment, we used an adaptation of the Few-Shot approach explained in (Brown et al., 2020), in order to influence the generated responses with the teacher’s style. For choosing the examples provided in the prompt, we used sentence embeddings generated with the gtr-t5-large-1-epoch model in hugging face<sup>3</sup>. An embedding was generated for each of the utterances in the training set partition. For generating a new response, the previous utterances are

<sup>2</sup><https://huggingface.co/tloen/alpaca-lora-7b>

<sup>3</sup><https://huggingface.co/cohere-io/gtr-t5-large-1-epoch>

```

### INSTRUCTION: You are an English teacher helping a
student on their learning process. Given a conversation,
generate the next teacher response.
The following are 3 examples of teacher's responses in
similar conversations you can use as reference:
"OK great, correct! Did i say exactly when?"
"Remember to log out if you can"
"You've probably heard about it"

### CONVERSATION:
teacher: could you finish the sentence, please?
student: I'll stay at home
teacher: great!
teacher: what about
teacher: If I ____ in the USA now, I ____ eat cheeseburgers
for breakfast!
teacher: Silly example, I know, just for the grammar
student: Have been/ will
teacher: How about 'If I were in the USA now, I would
eat...'

### RESPONSE:
teacher:

```

Added examples

Figure 4: Few-Shot prompt used with Alpaca LoRA applied to the example with id train\_1504 of the training set.

converted into an embedding and the three most similar conversations are selected from the training set using the k-Nearest Neighbors technique. The three responses of these selected examples are then added to the prompt, as can be seen in Fig. 4.

### 3.2 Fine-tuning pretrained Large Language Models

Pretrained LLMs tend to perform well in various tasks due to scaling up of model size, dataset size diversity, and length of training (Brown et al., 2020). However, using these models only with prompting techniques does not allow adapting to a target domain or target task, nor fully leveraging the potential of the training dataset.

Fine-tuning is the process of updating the weights of a pre-trained model by using a domain specific dataset in the training step. This technique tends to obtain strong performance in many benchmarks (Brown et al., 2020). However, it can be computationally very costly as all parameters of the LLM need to be updated. This is a major constraint, and sets a limit to the size of the models that we are able to fine-tune.

For this experiments we used the CluserUY infrastructure (Nesmachnow and Iturriaga, 2019), which has two servers using NVIDIA A100 GPUs and 28 servers using NVIDIA P100 GPUs.

#### 3.2.1 Experiments updating all the weights

DialoGPT is a transformer conversational model developed by Microsoft. It is based on the architecture of GPT2, which is known for its effectiveness in generating coherent and con-

textually relevant text. The specific implementation of DialoGPT used in our study is microsoft/dialogpt-large, which has 762 million parameters (Zhang et al., 2020b).

During training, DialoGPT was exposed to a vast amount of data, including 147 million conversation-like exchanges. These exchanges were extracted from Reddit comment chains spanning from 2005 through 2017. This diverse and extensive training data helped DialoGPT learn to generate responses that resemble human-like conversations.

As mentioned in (Zhang et al., 2020b), the human evaluation results demonstrate that the responses generated by DialoGPT exhibit a level of quality comparable to human responses in a single-turn conversation Turing test. Considering that the competition assesses the similarity to human responses as a metric, leveraging DialoGPT's performance has the potential to enhance the metrics of our model results.

It is important to note that in our study, we trained DialoGPT without specifically optimizing its architecture or training process. Our primary intention was to assess whether a conversational model like DialoGPT could achieve comparable performance to other existing models.

#### 3.2.2 Experiments using Low-Rank Adaptation

The high computational requirements for fine-tuning big LLMs, such as LLaMA 7b, posed a significant challenge even with access to the ClusterUY infrastructure. The process is not only computationally costly but also time consuming, which makes the task of training and testing various fine-tuned models with different base models or prompting techniques impractical. To overcome these restrictions, we opted to use Low-Rank Adaptation (LoRA) (Hu et al., 2021) for fine-tuning the bigger models.

LoRA is a method for fine-tuning models which aims to reduce GPU memory requirement by freezing the pretrained model weights and injecting trainable rank decomposition matrices into each layer of the Transformer architecture, reducing the amount of trainable weights. This method not only reduces computing and time requirements, but also space requirements because only the rank decomposition matrices need to be stored, which have much less parameters than the original matrices.

For example, suppose  $W \in \mathbb{M}_{m \times n}$  is a weight matrix and  $\Delta W \in \mathbb{M}_{m \times n}$  is the weight update

we want to learn. As shown in (Raschka, 2023), instead of learning  $\Delta W$ , we can decompose it into two smaller matrices:  $\Delta W = W_m W_n$ , where  $W_m \in \mathbb{M}_{m \times r}$ ,  $W_n \in \mathbb{M}_{r \times n}$  and  $r$  is a small number called rank. Keeping the original weights frozen and only training these new matrices results in reducing the amount of trainable parameters from  $m * n$  to  $m * r + r * n$ . After training, the new parameters are obtained by doing:  $W + W_m W_n$ .

Using the LoRA method, we trained fine-tuned versions of OPT 2.7b (Zhang et al., 2022), Bloom 3b (Scao et al., 2022) and LLaMA 7b (Touvron et al., 2023). For generating the dataset necessary to train all of these models, we adapted the training set in the following manner: The utterances and the response were joined into a string with a classical chat format, where every teacher intervention starts in a new line with “*teacher:*” and every student intervention starts in a new line with “*student:*”.

The configuration used for fine-tuning these models with LoRA involved a rank of 16, a scaling factor for the weight matrices of 32, and a dropout probability for the LoRA layers of 0.05. The training process employed the AdamW optimizer, with a total of 200 training steps, a learning rate of  $2 \times 10^{-4}$ , and a batch size of 4.

### 3.3 Preprocessing and Fine-Tuning

#### 3.3.1 Preprocessing technique

Upon analyzing the results during the development phase, we observed a recurring issue where the model became confused when attempting to continue the conversation from the teacher’s perspective after the same teacher had spoken. This discrepancy stemmed from the dataset’s structure, as it did not adhere to the conventional alternation of turns between speakers, which the models typically expect.

Consequently, even when explicitly specifying that the model should respond as a teacher in the prompt or using an input format like “Teacher: <Sentence-Before-Response>\n Teacher:”, the models consistently generated responses from the student’s standpoint. This posed a significant challenge not only during the model’s training phase, where it could become perplexed by the corpus structure, but also during the validation process.

To address this issue, we implemented two modifications:

**Corpus Modification:** We adjusted the corpus by introducing a structural change. Whenever two

consecutive conversations appeared in the original corpus, we combined them into a single utterance separated by a period. This alteration aimed to create longer utterances that would help the model distinguish between student and teacher interactions.

**Test-time Adjustment:** During testing, if the last utterance belonged to a teacher, we introduced an auxiliary phrase into the corpus. This additional phrase was carefully crafted to avoid introducing new information to the conversation, ensuring it did not hinder the teacher’s train of thought. We opted for the phrase “Student: I see\n,” a common expression used in the corpus and everyday conversations to convey active listening and encourage the other person to continue speaking.

By employing these preprocessing techniques, we sought to improve the model’s performance by aligning its responses more closely with the intended teacher’s perspective while overcoming the challenges posed by the dataset’s structure.

#### 3.3.2 Fine-Tuned model using the preprocessing technique

The model in which we used this ad-hoc technique was opt-2.7b (Zhang et al., 2022). OPT, developed by Meta, is a decoder-only language model closely related to GPT-3. It has been predominantly pretrained on English text, supplemented with a small amount of non-English data obtained from CommonCrawl. The model’s pretraining process employed a causal language modeling (CLM) objective, similar to other models in its family. Evaluation of OPT aligns with the prompts and experimental setup used for GPT-3 (Brown et al., 2020).

The decision to employ OPT in this study was motivated by the aim of exploring an alternative that offers both variety and considerable power. However, it is crucial to acknowledge and address the limitations of this model. Meta AI’s model card highlights that OPT’s training data consists of unfiltered internet content, resulting in a significant bias embedded within the model.

The configuration used for fine-tuning this model was the AdamW optimizer, a learning rate of 0.001 and a batch size of 4.

#### 3.4 Combining prompting techniques with fine-tuning

After experimenting with prompt-based and fine-tuning approaches, a natural evolution was to look for ways to combine both of these techniques. Our



first approach was to fine-tune the model LLaMA 7b with LoRA using the already explained few-shot method. In the same way as before, the three most similar responses in the training set with respect to the reference response were chosen to be added to the context. We took into consideration that responses from different partitions of the same conversation should not be considered for this selection. We expected that during fine-tuning, some patterns that could exist between the similar responses and the expected response could be learned.

Recent works like (Wei et al., 2023) showed that adding intermediate reasoning steps that lead to the final answer for a problem improves the ability of LLMs to perform complex reasoning. Inspired on this work, we designed a different solution that tries to combine intermediate reasoning and fine-tuning.

The training set was modified to include some characteristics of the response. Initially, two new features were added. A binary feature that is set to 1 if the response has a question mark, and a multiclass feature that is composed of 28 emotions taken from (Demszky et al., 2020), such as anger, approval, curiosity, disapproval, neutral and others. To obtain the second feature for every example in the training set, the EmoRoBERTa model was used (Kamath et al., 2022). This model classifies text into the 28 emotions already mentioned.

Then, a dataset for fine-tuning was constructed. Each example of the dataset is a string composed of three sections: *Conversation*, where the utterances are presented in a classical chat format, *Reflection*, which is constructed using the already mentioned features, and *Response*, which has the reference response.

The *Reflection* section is a sentence with two parts: The first part indicates the expected emotion of the response and the second part, which is optional, indicates if the expected response is a question. For example, an example classified as “*Curiosity*” and that is a question would have the reflection: “*My response should show curiosity and should be a question*”. A complete example can be seen in figure 5.

Using this dataset, we fine-tuned LLaMA 7b with the already mentioned LoRA technique. Given a new conversation, the model is capable of generating a complete reflection and response. The reflection is discarded to get the final response.

A second version was created using a new feature that classifies the response length in short, nor-

```

### CONVERSATION:
student: I understood the stuffs of present continuous when I was study online.
student: I understood the stuffs of present continuous when I was studying online.
teacher: OK thanks - the second one = correct goo

### REFLECTION:
My response should show caring and should be a question.

### RESPONSE:
teacher: Try one more if you can OK?

```

Figure 5: Example of the prompt used for the reflection approach dataset.

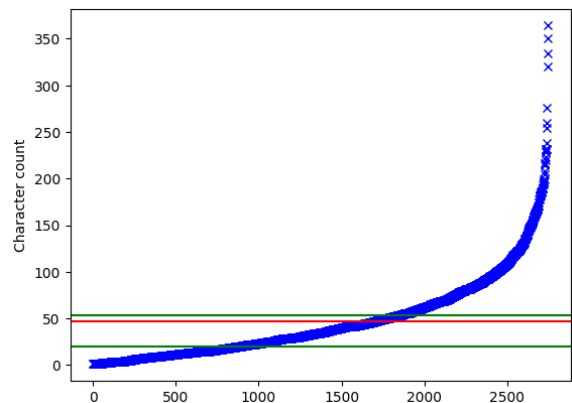


Figure 6: Character count per example in the training set, in ascending order. The green lines indicate the thresholds of each class, and the red line indicates the average.

mal or long. A response is considered short if it has 20 characters or less and long if it has 53 characters or more. These numbers were selected in order to divide the dataset in the most balanced way (approximately 1/3rd for each class) as can be appreciated in Fig. 6. The reflection sentence was changed to include this information.

## 4 Results

Given that this work is framed in the context of the BEA 2023 shared task, and the development and test sets gold responses were not released until after the competition finished, we created our own internal split of the training set in 80% for training and 20% for internal validation. We will present the results of all our experiments against this internal validation data, which we call the internal validation phase. For the development and test sets, we will only present the results of the systems submitted to the competition.

A problem with this internal split, as already explained in the data analysis section, is that it includes some repeated utterances across the training and validation sets, due to the overlapping that oc-

Experiment	BERTScore			DialogRPT		
	Precision	Recall	F1 Score	updown	human_vs_rand	human_vs_mach
Finetuning (LoRA) Bloom 3b	0.840	0.838	0.838	0.495	<b>0.912</b>	0.985
Finetuning (LoRA) Llama 7b + Reflection	0.808	0.840	0.823	0.463	0.881	0.995
Alpaca LoRA	0.832	0.829	0.830	0.489	0.841	0.986
Alpaca LoRA + Few Shot	0.836	0.836	0.836	0.480	0.820	0.989
Finetuning (LoRA) Llama 7b + Few Shot	0.802	0.839	0.819	0.465	0.871	<b>0.997</b>
Finetuning (LoRA) opt 2.7b	0.841	0.832	0.836	0.478	0.748	0.966
Finetuning opt 2.7b	0.847	<b>0.842</b>	0.844	0.474	0.673	0.981
Finetuning (LoRA) Llama 7b	0.854	0.841	<b>0.847</b>	0.473	0.642	0.965
Finetuning (LoRA) Llama 7b + Reflection with length	0.850	0.831	0.840	0.465	0.595	0.985
Finetuning DialogGPT Large	0.700	0.667	0.682	0.462	0.592	0.959
Baseline 1: Always reply "Hello"	<b>0.861</b>	0.805	0.832	<b>0.524</b>	0.305	0.952
Baseline 2: Always reply "Cucumber"	0.723	0.810	0.764	0.503	0.360	0.992

TABLE 2: Internal validation results.

curs in some of the training set tuples. This may influence the results during evaluation, but we decided to keep it this way so as not to significantly reduce the training set partition.

Two evaluation metrics are used in all phases, following the indications given in the official website of the shared task<sup>4</sup>: One of them is BERTScore (Zhang et al., 2020a), which produces precision, recall, and F1 scores by comparing words in the generated response with respect to the reference response using cosine similarity. The other one is DialogRPT (Gao et al., 2020), which evaluates the generated response taking into account the utterances given as context. The specific DialogRPT metrics used are updown, human\_vs\_rand, human\_vs\_machine and final (average and best).

#### 4.1 Internal evaluation

Due to the fact that both metrics have multiple hyperparameters that can be tuned differently, the configuration used during this internal phase does not align exactly with the one used in the competition. For the BERTScore metric, roberta-large is used as the base model and idf weighting is not used. Meanwhile, for DialogRPT, the context used are the utterances concatenated in a classical chat format and the hypothesis is the generated response.

Trying out different configurations for DialogRPT, we found out that the definition of the context to be used has a big influence on the results obtained. As no information was provided on how the context was going to be defined in the development and evaluation phases, we made our own definition and used it consistently during all our internal evaluations.

<sup>4</sup><https://sig-edu.org/sharedtask/2023#evaluation>

The results obtained during the internal evaluation for all the described experiments can be observed in Table 2. Besides all the methods described, we include two very simple methods that serve as baselines to compare with. In both cases the baseline systems generate the same response to all contexts. One baseline always replies "Hello", and the other always replies "Cucumber", so as to consider a more likely and a more unlikely case.

#### 4.2 Development and evaluation phases

For the development phase, we decided to submit the LoRA fine-tuning of the model LLaMA 7b, which had the best F1 score in the internal phase, the model Alpaca LoRA with the Few-Shot technique for the prompt, and the fine-tuned version of DialogGPT. We chose to submit these models because each of them uses a different approach: fine-tuning with LoRA, a prompting technique, and fine-tuning updating all the weights, respectively. It is important to mention that not all the experiments were completed when the deadline for this phase occurred.

Due to an error in the calculation of BERTScore on CodaLab<sup>5</sup>, the results obtained in the development phase were not correct. This influenced our decisions of what models to send to the evaluation phase, given that our internal evaluations did not seem to correlate with these obtained results. The corrected results were later published, and can be seen in Table 3.

Considering that the Alpaca LoRA with Few-Shot approach was the one that yielded the best results in the development phase, we decided to also submit it in the evaluation phase. Two new approaches were also submitted: the LoRA fine-

<sup>5</sup><https://codalab.lisn.upsaclay.fr/>

Experiment	BERTScore			DialogRPT				
	Precision	Recall	F1 Score	updown	human_vs_rand	human_vs_mach	final (avg)	final (best)
Finetuning (LoRA) Llama 7b	<b>0.72</b>	<b>0.70</b>	<b>0.71</b>	0.36	0.94	0.98	0.32	0.67
Alpaca LoRA + Few Shot	0.68	0.69	0.68	<b>0.37</b>	<b>0.95</b>	<b>0.98</b>	<b>0.33</b>	<b>0.72</b>
Finetuning DialoGPT Large	0.70	0.67	0.68	0.35	0.92	0.98	0.30	0.68

TABLE 3: Development phase results.

tuning of LLaMA 7b with reflection in the prompt, and the fine-tuning of OPT 2.7b with preprocessing. Table 4 shows the results obtained for this phase, evaluated over the test set.

### 4.3 Observations

We observed that fine-tuning a model updating all the weights does not show significant differences in comparison to using the LoRA technique. On a separate note, the results reveal that fine-tuned models seem to improve the BERTScore results over prompting techniques, but the opposite seems to happen with DialogRPT metrics. The experiments that try to combine both techniques tend to show competitive results across all metrics.

Another observation that derives from the internal results (Table 2), is that the "Hello" baseline approach not only yields good results in the majority of the metrics, but is also the best in BERTScore precision and DialogRPT updown. This seems to indicate that these metrics (at least with our configuration) may not fully capture or accurately correlate with human judgement.

## 5 Conclusions

We presented the experiments we performed for the BEA 2023 shared task on generating teacher responses in educational dialogues. Our methods use the latest open source LLMs in a variety of scenarios and incorporating some fine-tuning and targeted prompting strategies for improving the performance.

The experiment that yielded best results in the development phase was the model Alpaca LoRA with a Few-Shot prompting technique, which ranked third. However, in the evaluation phase, the Fine-Tuning version of OPT 2.7b with preprocessing ended up performing better than the previous one, and ranked fourth in this phase.

### 5.1 Areas of Improvement

Throughout the competition, several areas were identified where improvements could have enhanced the performance of our chatbot model.

On the one hand, further fine-tuning of the model's parameters could have been explored to optimize its performance. By carefully tuning hyperparameters, we could have potentially achieved better results in terms of response quality and coherence. Additionally, despite training our models using high-performance GPUs (e.g., A100 and P100), we faced limitations in testing models with more than 10 billion parameters. Given the advancements in model architectures, exploring larger models could have yielded further improvements in chatbot performance. Overcoming hardware limitations and resource constraints would open avenues for investigating more powerful models in future iterations. Moreover, to resource and time constraints, our models could not be trained for different number of epochs. Longer training durations are often beneficial for improving model performance. Given more resources and time, training the models for multiple epochs could have yielded better results.

On the other hand, one challenge encountered during the competition was data leakage between the internal validation set and the training set. This issue, arising from the training dataset, hindered the models' ability to accurately improve their performance without overfitting. A more carefully curated validation set, separate from the training data, would have provided a more reliable evaluation metric. Furthermore, regarding the evaluation metrics, BERTScore and DialogRPT, we observed questionable scores when comparing our model's performance against a baseline of answering "hello" for every prompt. The BERTScore showed unexpectedly high scores for this baseline, while DialogRPT correctly penalized such responses. On top of that, another baseline that responded with a fixed word "cucumber" consistently scored poorly, which aligns with our expectations. Careful consideration and refinement of our evaluation metrics are necessary to ensure their reliability and alignment with the desired behavior of chatbot models.

Experiment	BERTScore			DialogRPT				
	Precision	Recall	F1 Score	updown	human_vs_rand	human_vs_mach	final (avg)	final (best)
Finetuning (LoRA) Llama 7b + Reflection	0.73	<b>0.71</b>	<b>0.72</b>	0.37	<b>0.94</b>	<b>0.98</b>	0.33	0.64
Finetuning opt 2.7b	<b>0.74</b>	0.68	0.71	0.38	0.90	0.96	<b>0.35</b>	0.65
Alpaca LoRA + Few Shot	0.72	0.68	0.70	0.37	0.91	0.96	0.34	<b>0.68</b>

TABLE 4: Evaluation phase results.

## 5.2 Ethical limitations

It is essential to address the ethical limitations observed our fine-tuned OPT model, ranked 4th in the competition. The model card provided by Meta AI highlighted that the training data used for their model consisted of unfiltered internet content, leading to the presence of significant biases within the model. These ethical considerations raise concerns regarding fairness, inclusivity, and potential biases in the responses generated by the model. Further research and development in addressing these limitations are imperative to ensure the responsible and unbiased deployment of chatbot models.

## 5.3 Final thoughts

In conclusion, while our chatbot models showcased promising performance in the competition, there are areas for improvement and important ethical considerations to be addressed. By focusing on adjusting model parameters, handling specific tokens, increasing training duration, improving validation sets as well as their preprocessing, and exploring larger models, future iterations of chatbot models can achieve even greater performance and ensure ethical deployment.

## Acknowledgements

Some experiments presented in this paper were carried out using ClusterUY (site: <https://cluster.uu>).

## References

Serge Bibauw, Wim Van den Noortgate, Thomas François, and Piet Desmet. 2022. Dialogue systems for language learning: a meta-analysis. *Language Learning & Technology*, 26(1).

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc."

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, S. Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S.

Chen, Kathleen A. Creel, Jared Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren E. Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas F. Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, O. Khat-tab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir P. Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avnika Narayan, Deepak Narayanan, Benjamin Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, J. F. Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Robert Reich, Hongyu Ren, Frieda Rong, Yusuf H. Roohani, Camilo Ruiz, Jack Ryan, Christopher R'e, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishna Parasuram Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei A. Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2021. [On the opportunities and risks of foundation models](#). *ArXiv*.

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](#).

Andrew Caines, Helen Yannakoudakis, Helena Edmondson, Helen Allen, Pascual Pérez-Paredes, Bill Byrne, and Paula Buttery. 2020. The teacher-student chat-room corpus. *arXiv preprint arXiv:2011.07109*.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A Dataset of Fine-Grained Emotions. In *58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

- Xiang Gao, Yizhe Zhang, Michel Galley, Chris Brockett, and Bill Dolan. 2020. [Dialogue response ranking training with large-scale human feedback data](#).
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. [Lora: Low-rank adaptation of large language models](#).
- Rohan Kamath, Arpan Ghoshal, Sivaraman Eswaran, and Prasad Honnavalli. 2022. [Emoroberta: An enhanced emotion detection model using roberta](#). *SSRN Electronic Journal*.
- Sergio Nesmachnow and Santiago Iturriaga. 2019. Cluster-uy: Collaborative scientific high performance computing in uruguay. In *Supercomputing*, pages 188–202, Cham. Springer International Publishing.
- Sebastian Raschka. 2023. [Parameter-efficient llm fine-tuning with low-rank adaptation \(lora\)](#).
- Teven Le Scao, Angela Fan, Christopher Akiki, Elie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. *arXiv preprint arXiv:2211.05100*.
- Anaïs Tack, Ekaterina Kochmar, Zheng Yuan, Serge Bibauw, and Chris Piech. 2023. The BEA 2023 Shared Task on Generating AI Teacher Responses in Educational Dialogues. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications*, page to appear, Toronto, Canada. Association for Computational Linguistics.
- Anaïs Tack and Chris Piech. 2022. The ai teacher test: Measuring the pedagogical ability of blender and gpt-3 in educational dialogues. *ArXiv*, abs/2205.07540.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. [https://github.com/tatsu-lab/stanford\\_alpaca](https://github.com/tatsu-lab/stanford_alpaca).
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. [Llama: Open and efficient foundation language models](#).
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. [Chain-of-thought prompting elicits reasoning in large language models](#).
- S. Wollny, J. Schneider, D. Di Mitri, J. Weidlich, M. Ritterberger, and H. Drachsler. 2021. [Are we there yet? - a systematic literature review on chatbots in education](#). *Frontiers in artificial intelligence*.
- Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. [Opt: Open pre-trained transformer language models](#).
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020a. [Bertscore: Evaluating text generation with bert](#).
- Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020b. [Dialogpt: Large-scale generative pre-training for conversational response generation](#).

# Empowering Conversational Agents using Semantic In-Context Learning

Amin Omidvar, Aijun An

Department of Electrical Engineering and Computer Science, York University, Canada  
omidvar@yorku.ca, ann@yorku.ca

## Abstract

Language models are one of the biggest game changers in downstream NLP applications, especially in conversational agents. In spite of their awesome capabilities to generate responses to solve the inquiries, there are still some big challenges to using them. One challenge is how to enable the LLMs to use the private internal data to solve inquiries. And secondly, how to keep the LLMs updated with newly incoming data without the burden of fine-tuning as it is not only expensive but also not an available option for some commercial LLMs, such as ChatGPT. In this work, we propose Semantic In-Context Learning (S-ICL) to address the aforementioned challenges. Our proposed approach participated in the BEA 2023 shared task<sup>1</sup> and ended up achieving the fourth place in both the development and evaluation phases.

## 1 Introduction

Conversational agents are one of the most important applications of NLP. If implemented successfully, they can bring tremendous benefits for both organizations and clients, such as improving the efficiency of customer service in terms of support and availability of the services.

With the emergence of powerful large language models (LLMs) such as ChatGPT, there is a lot of interest in leveraging LLMs to develop AI agents. Even though LLMs are capable of answering a broad spectrum of questions, there are still two major bottlenecks for using them as an AI assistant.

First, each organization has some valuable internal knowledge such as FAQs, policies, regulations, etc. that can or should be used to resolve incoming inquiries. However, the LLMs are trained based on public datasets and may not be aware of private knowledge sources that could help them to resolve incoming inquiries more accurately.

<sup>1</sup><https://sig-edu.org/sharedtask/2023>  
Our username and team's are amino and aitis, respectively.

Secondly, fine-tuning these LLMs on the organization's internal data is not an easy task due to factors such as the size of the LLMs, cost of training, frequent updates in the internal data, and data privacy. For example, in news media, news articles are published every day that LLMs are not aware of them. If the news media decides to use an LLM as an agent, the agent would be unable to provide users with information about current events or answer their questions about what is happening now. On top of that, the fine-tuning option is not available for certain LLMs (e.g., ChatGPT with the GPT-3.5-turbo engine).

One possible solution to the mentioned problems is In-Context Learning (ICL), as it can enable the LLMs to perform well on the tasks or data that they have never seen before (Brown et al., 2020). In ICL, a prompt containing an instruction, few labeled samples, and an unlabeled sample is given to the LLM. Then, the LLM would be able to label the unlabeled sample without the need for any gradient-based training (Liu et al., 2022).

However, it is infeasible to show all the available samples to the LLM due to the high cost of computation. Also, previous research shows that the format of the prompt, the selection of samples, the number, order, and structure of samples could have not only significant but also unforeseeable effects on LLMs' performance (Min et al., 2022; Sanh et al., 2021; Wei et al., 2023; Liu et al., 2022).

To solve the aforementioned problems, we propose Semantic In-Context Learning (S-ICL) which utilizes a semantic search engine (i.e., an SBERT model (Reimers and Gurevych, 2019)) and an LLM (i.e., ChatGPT with the gpt-3.5-turbo engine) to build a conversational agent. This agent not only benefits from the knowledge of an LLM but also utilizes available private knowledge sources to provide the correct answer to the inquiries. We also propose a flexible architecture that allows experts to apply and compare different approaches for prompt

engineering.

The proposed model is developed and participated in the BEA 2023 Shared task (Tack et al., 2023). However, the proposed model is flexible, and the agent can be used in other domains such as news media, customer service, and more.

The rest of the paper is as follows. In Section 2, we describe the proposed architecture along with its components. In section 3, we compare different configurations of the proposed model on the created test set, and we also evaluate the model on the competitor's data. Finally, this paper is wrapped up with the conclusion in section 4.

## 2 Proposed Model

In this section, we present our proposed approach for generating a response to the inquiry. Our proposed approach uses semantic search (Reimers and Gurevych, 2019) to enable the agent to utilize private domain data. It also uses a large language model not only to provide higher quality answers but also to enable the agent to answer questions that are significantly different from past questions and answers in the private domain data.

### 2.1 Overview

As shown in Figure 1, the proposed architecture consists of five main components: Data pre-processor, Embedder, Retriever, Prompt builder, and Answer generator. The first three components are related to the semantic search part of the architecture, while the other two are related to the language model.

### 2.2 Data pre-processor

The data pre-processor receives utterances in JSON format containing a context and a query (i.e., the last utterance). It extracts and transforms the JSON file into the followings:

**Concatenation:** it's a textual concatenation of all the utterances made by a student and a teacher. The main purpose of transforming data into this format is to enable its use in the semantic search part of the architecture.

**Sample:** It's a conversational flow between the student and the teacher. Based on who wrote the utterance, either "Teacher: " or "Student: " would be appended in the beginning of the utterance. This format is being used by the prompt builder component as it is more appropriate to be used by the language model.

### 2.3 Embedder

In this section, we use a state-of-the-art transformer encoder model to convert the concatenation format, which is built in the data pre-processor part, into the embedding representation. We use the pre-trained model "multi-qa-mpnet-base-dot-v1" to generate embeddings as it has the highest performance in the Hugging Face benchmark<sup>2</sup>. The tokenizer first tokenizes the input text, and then the transformer encoder model infers an embedding vector with a size of 768 for each token of the input text. The embedding vector of the CLS token in the last layer is considered the embedding representation of the whole input text.

### 2.4 Retriever

The Retriever is responsible for finding the most similar records that exist in the training data to the incoming context. It calculates the cosine similarity between the embedding vector of the context and each embedding vector in the training set. Then, the results would be sorted in descending order based on the cosine similarity score, and the top N results would be passed on to the next step.

This process could be significantly sped up on large datasets by using approximate K-nearest neighbor methods, such as Facebook AI Similarity Search (Faiss) (Johnson et al., 2019). However, due to the small size of our data, we don't need to use any approximate K-NN methods.

### 2.5 Prompt builder

The prompt builder component creates a prompt based on the selected prompt building approach. Figure 2 shows the structure of the prompt which consists of the following components in order:

**Command:** It's a first component of the prompt that informs the language model of what is expected to be done.

**Sample(s):** The retrieved sample(s) from the training set are included to assist the language model in answering the inquiry. This part of the prompt is optional because the number of samples to be used depends on the selected approach.

**Inquiry:** It contains the last utterance along with the previous utterances (i.e., Context) given to the system.

The command part of the prompt is written by humans, while the other parts are generated auto-

<sup>2</sup>[https://www.sbert.net/docs/pretrained\\_models.html](https://www.sbert.net/docs/pretrained_models.html)

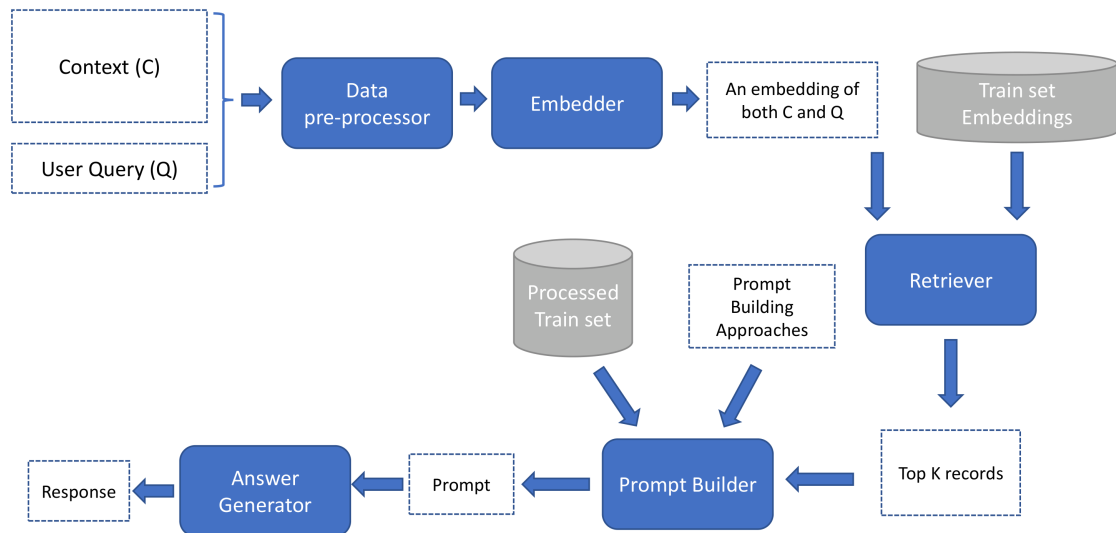


Figure 1: The proposed architecture for the conversational agent uses both semantic search and a large language model.

matically depending on the chosen prompt building approach. So far, four prompt building approaches have been designed, but more could be defined to further improve the agent’s performance or adapt it better to different domains of data, such as news.

## 2.6 Answer Generator

A large language model is used in this part of the architecture. In our experiment, we use ChatGPT<sup>3</sup> API using gpt-3.5-turbo engine. A prompt created in the previous stage would be sent to the language model, and the response would be returned to the end user. To make the result reproducible, we set the temperature value to zero.

In this way, the language model can not only use its knowledge but also have access to the relevant past responses from the private domain knowledge to answer the question. Another advantage is that there is no need to fine-tune the large language model on private internal data, which may not be an option for many models, such as ChatGPT.

## 3 Experiment

This section has three subsections. In the first subsection, we introduce the dataset used, split the train portion of the data into our created train and test sets, and show how the pre-processing has been done. In the second subsection, we conduct experiments on the proposed architecture using the created test set (i.e., selected from the original training set) and compare the accuracy of the model using

different prompt building approaches. In the third subsection, we will demonstrate the model’s performance on the development and testing sets of the competition data.

### 3.1 Data

The data consists of the conversation between a student and a teacher provided by (Caines et al., 2020). The sizes of the provided data and their release dates in the competition are shown in Table 1. We transform the training set using the pre-processor component (subsection 2.2). Then, we use the embedder component (subsection 2.3) to convert the concatenation of the utterances into their embedding representations (i.e., Train set embedding in Figure 1).

Then, we split the train set into customized train and test sets with sizes of 2647 and 100, respectively. We use the customized train and test sets to compare the different prompt generation approaches in subsection 3.2. Since some of the records in the training set have similar utterances (i.e., they overlap), we select the test data in a way that none of the test conversations can be answered directly from the conversations in the training set (i.e., there is no overlap between the utterances of the train and test sets).

### 3.2 Evaluation of different approaches

We use five different approaches to provide the response to the incoming inquiry. In the first approach, we only use the semantic search. That

<sup>3</sup><https://openai.com/blog/chatgpt>



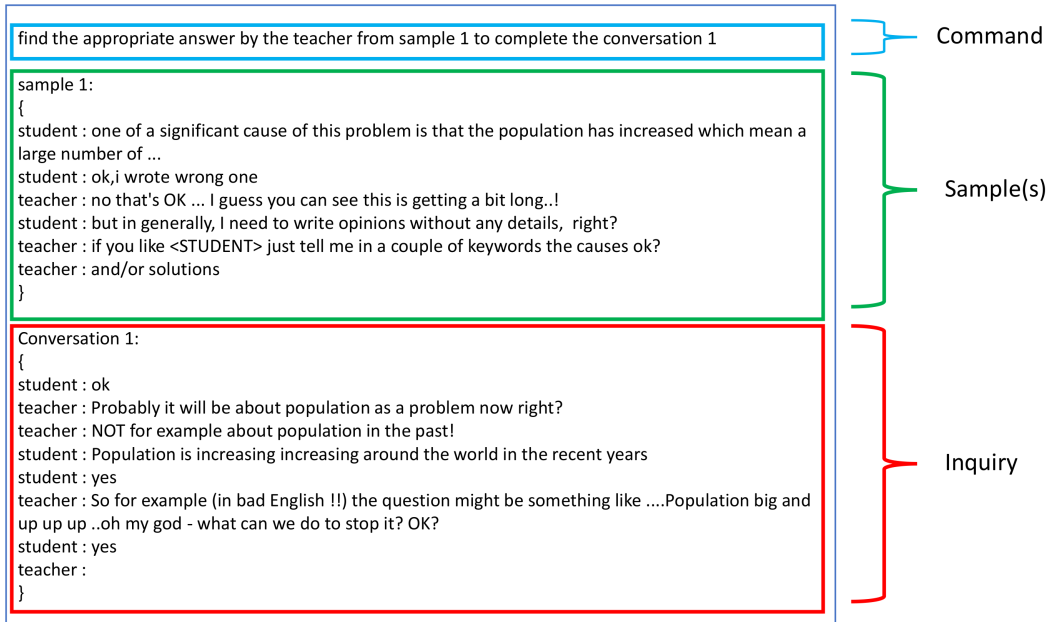


Figure 2: An example of a prompt structure with a single sample.

Set	Size	Release date
Train	2747	March 24, 2023
Dev	305	March 24, 2023
Test	273	May 1, 2023

Table 1: The statistics of the TSCC dataset.

means the last utterance of the most similar retrieved sample is chosen as a response. Next, we are curious to see how good the language model is in completing the conversation without using any samples. The command we use is "Complete the following conversation by giving an appropriate answer by the teacher". However, for the third approach, we ask the language model to "Find the appropriate answer by the teacher from sample 1 to complete the conversation 1". The provided sample, which has the ID "train\_0063", was chosen by us from the training set and has been used for all inquiries.

During the experiments, we observed that ChatGPT tends to generate longer responses than the ground truths. However, we discovered that by formulating our prompt command in a certain way (i.e., find the appropriate answer by the teacher from sample ...), ChatGPT can produce more concise and shorter responses. Therefore, we decided to write the command part of our prompt in this way. We also observed that for some inquiries, ChatGPT mentions "teacher :" in its response, so we wrote a rule to remove it.

The fourth approach includes the top 3 most similar samples in the prompt and the command is "Find the appropriate answer by the teacher from sample 1, sample 2 and sample 3 to complete the conversation 1". And the last approach is similar to the third one but instead of using the curated sample, the most similar sample from the training set is being used. The last two approaches are based on S-ICL.

The results of the above approaches on the created test dataset are shown in Table 2 in terms of BERT Score (Zhang et al., 2019) and DialogRPT (Gao et al., 2020). In Table 2, P, R, F, U, HvR, HvM stand for precision, recall, f1-score, updown (the probability that a response receives upvotes), human vs random (the probability that the response is relevant to the given context), human vs machine (the probability that the response was written by a human rather than generated by a machine), respectively. The first three measures belong to BERTScore (Zhang et al., 2019), and the rest of them belong to DialogRPT (Gao et al., 2020). We use "roberta-large" model<sup>4</sup> for the BERTScore as we do not know which model the competition is using. We then compare the generated responses with their ground-truths using BERTScore in terms of precision, recall, and f1-score. Each of the first three measures of DialogRPT (i.e., U, HvR, and HvM)<sup>5</sup> has its own pre-trained model. Each model

<sup>4</sup><https://huggingface.co/roberta-large>

<sup>5</sup><https://github.com/golsun/DialogRPT>

Approach	BERTScore			DialogRPT				
	P	R	F	U	HvR	HvM	best	avg
1	0.824	0.823	0.823	0.433	0.789	0.983	0.999	0.735
2	0.835	<b>0.837</b>	0.836	0.490	<b>0.922</b>	<b>0.999</b>	<b>0.999</b>	<b>0.804</b>
3	<b>0.839</b>	0.836	<b>0.837</b>	0.464	0.877	0.997	0.998	0.779
4	0.828	0.832	0.830	<b>0.574</b>	0.767	0.998	0.999	0.780
5	0.835	0.831	0.833	0.481	0.839	0.997	0.999	0.773

Table 2: The comparison between different approaches used on the created test set in terms of BERT Score.

receives the generated responses and their corresponding contexts (i.e., the previous utterances of each conversation) to calculate a score.

Interestingly, the model that uses the fixed sample for all the inquiries (third approach) gained the best BERTscore in terms of f1-score. This observation is inline with the results of other studies such as (Min et al., 2022) that they concluded replacing the sample labels randomly would barely hurts the performance of the LLMs. In terms of DialogRPT, the second approach gained the best results. However, when we examined the generated answers, we found out the answers of the fifth approach are both more reasonable and preferable in comparison with the other approaches.

### 3.3 BEA Workshop’s evaluation

Our proposed approach ranked fourth both in development and evaluation phases. We used our third approach (using the fixed sample) for the development phase as we noticed the majority of utterances in development data have overlap with the training set. If we use either the fifth or fourth approach, the model would recognize the similarity between the sample and the conversation and produce a response so similar to the existing utterance in the sample that it would inflate the performance of the system. However, we discovered that the test data is different in a way that none of its conversations could have their responses directly obtained from any utterances in either the training or development sets. Therefore, for the evaluation set, we used the fifth approach. Another reason that why we used the fifth approach in the evaluation phase is that the top three models would be evaluated by the human evaluators, and we already noticed in subsection 3.2 that the results of the fifth approach are more desirable from humans’ point of view.

The evaluation phase was started on May 1st and ended on May 5th. Due to an unprecedented emergency, we were unable to continue working

on the test data and our last submission was on May 1st. Our model ended up ranking fourth in the evaluation phase and could not pass to the human evaluation phase. However, we think that the proposed model has a high potential for improvement, especially if more efforts would be put on the prompt engineering part of the architecture.

## 4 Conclusion

We proposed a Semantic In-Context Learning (S-ICM) approach for conversational agents using the combination of a semantic search and a large language model (i.e., ChatGPT). We also implemented an architecture enabling users to apply and compare different approaches for prompt engineering. We applied our proposed method on the BEA 2023 shared task and our approach ended up ranking fourth in both the development and evaluation phases.

## Acknowledgements

This work is funded by Natural Science and Engineering Research Council of Canada (NSERC).

## References

- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Andrew Caines, Helen Yannakoudakis, Helena Edmondson, Helen Allen, Pascual Pérez-Paredes, Bill Byrne, and Paula Buttery. 2020. The teacher-student chat-room corpus. *arXiv preprint arXiv:2011.07109*.
- Xiang Gao, Yizhe Zhang, Michel Galley, Chris Brockett, and Bill Dolan. 2020. Dialogue response ranking training with large-scale human feedback data. *arXiv preprint arXiv:2009.06978*.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

- Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohata, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965.
- Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.
- Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.
- Anaïs Tack, Ekaterina Kochmar, Zheng Yuan, Serge Bibauw, and Chris Piech. 2023. The BEA 2023 Shared Task on Generating AI Teacher Responses in Educational Dialogues. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications*, page to appear, Toronto, Canada. Association for Computational Linguistics.
- Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. 2023. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

# NAISTeacher: A Prompt and Rerank Approach to Generating Teacher Utterances in Educational Dialogues

Justin Vasselli<sup>†,††</sup> Christopher Vasselli<sup>††</sup> Adam Nohejl<sup>†</sup> Taro Watanabe<sup>†</sup>

<sup>†</sup> Nara Institute of Science and Technology <sup>††</sup> Serpenti Sei

<sup>†</sup> {vasselli.justin\_ray.vk4, nohejl.adam.mt3, taro}@is.naist.jp  
<sup>††</sup> chris@serpentisei.com

## Abstract

This paper presents our approach to the BEA 2023 shared task of generating teacher responses in educational dialogues, using the Teacher-Student Chatroom Corpus. Our system prompts GPT-3.5-turbo to generate initial suggestions, which are then subjected to reranking. We explore multiple strategies for candidate generation, including prompting for multiple candidates and employing iterative few-shot prompts with negative examples. We aggregate all candidate responses and rerank them based on DialogRPT scores. To handle consecutive turns in the dialogue data, we divide the task of generating teacher utterances into two components: teacher replies to the student and teacher continuations of previously sent messages. Through our proposed methodology, our system achieved the top score on both automated metrics and human evaluation, surpassing the reference human teachers on the latter.

## 1 Introduction

The shared task for BEA2023 was to generate teacher utterances in an educational dialogue, specifically one between an English language learner and their language teacher (Tack et al., 2023).

The data was collected from one-on-one English lessons between real teachers and students conducted over a chat application. The data for the task consists of fragments of these dialogues, with the goal of predicting the next teacher utterance.

Inspired by a commonly used practice in machine translation (Och and Ney, 2002; Shen et al., 2004; Lee et al., 2021), our system generates multiple candidates and reranks them. Given the high level of fluency required for this task, we began with a pretrained language model (GPT-3.5-turbo) rather than training one from scratch.

An overview of the system is outlined in Figure 1. First the prompts for the prompt ensemble are cho-

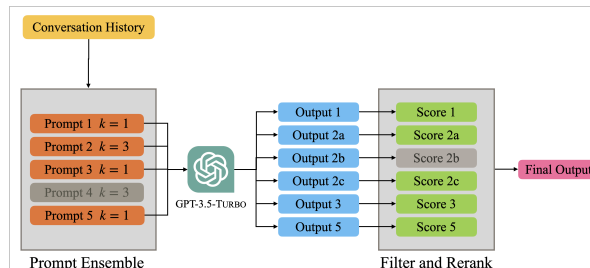


Figure 1: The NAISTeacher system overview. Prompts are chosen for the ensemble based on the role of the speaker of the final utterance of the conversation history.  $k = 1$  and  $k = 3$  refer to the number of candidates solicited by the prompt (one and three respectively).

sen based on the conversation history. The prompt ensemble is sent to GPT-3.5-turbo to generate a set of candidate responses. During a post-processing step some responses are flagged as inappropriate for referring to pronunciation, responding as a student, or containing profanity. These inappropriate responses are filtered out before reranking. The response chosen by the reranker is returned.

Our prompt ensemble consists of a mixture both zero and few shot prompts. For the few shot prompts, we experimented with several different ways of selecting the examples to provide, but the most effective was an iterative approach that, to our knowledge, is novel. This approach works in two steps: The first step is to generate the candidate teacher utterances for all conversations using the zero-shot prompt and score them. In the second step, the highest and lowest scoring responses are integrated into the prompt as positive and negative example responses.

To generate multiple candidates, we instructed the model to return multiple possible responses with a single prompt similar to Aumiller and Gertz (2022). We found this to be a very effective technique because it resulted in a diverse set of candidates to select from in the reranking stage.

We tested our methods on a held-out section

of the training data which contained reference responses using BERTScore (Zhang et al., 2019) and DialogRPT (Gao et al., 2020). We found the iterative few-shot approach increased the overall score of the data on both automated metrics. However, manual inspection of the output showed that this approach works better for some cases than others.

### Teacher response or teacher continuation?

As the data was collected from real-time chats between real people, the turns did not alternate consistently as they do in most dialogue systems. The task of generating a teacher continuation – that is, a teacher follow-up message when that same teacher was the last to have taken a turn in the conversational history – was particularly challenging. This phenomenon occurs in 38% of the training data, and 40% of the development data. As systems are trained to alternate turn taking, this led to many of the generated responses erroneously sounding like they were either uttered by the student, or showed a lack of awareness of the immediately preceding utterance.

Figure 2 shows one such conversation and an example of a problematic generated response. The teacher first asks the student to look at the text and think about if the word “startling” means obvious or surprising. The AI generated follow-up gives the answer away immediately (“Startling” means surprising), which overrides the original goal: to encourage the student to find the answer themselves.

To address this, we split the data into two tasks based on who spoke last: **teacher reply** for when the context ended with the student, and **teacher continuation** for when the context ended with the teacher. We evaluated each task separately during development to design a set of prompts for each.

## 2 Data

Although we explored augmenting with data from other sources, in the final system we only used the data made available for this shared task, the Teacher-Student Chatroom Corpus (Caines et al., 2020, 2022). The training data consists of 2747 partial conversations between a student learning English and their teacher. We split out a random sample of 275 conversations (10%) to use as validation data. Our experiments were carried out on this split because the development data supplied by the shared task did not contain reference answers.

Student	...provocative, and supplied +startling+ messages about psychology...
Student	Is it enough this piece of text to work on it? the word I don't know is startling
Teacher	ok thanks....'startling' does it mean: obvious or surprising do you think? is there a context in the text to help you?
Teacher	Great job using the keywords from the text! "Startling" means surprising. In this context, it seems like the text contains provocative and surprising messages about psychology. Can you think of any other words that might be related to the topic of psychology?

Figure 2: An example of a teacher continuation with the response to the zero-shot (1) prompt.

Besides a single run of our system on this data to submit to CodaLab, we did not use the 305 conversations in the supplied development dataset. The test data contained 273 conversations.

## 3 System

Our system generates a number of responses by prompting GPT-3.5-turbo using the default parameters, and chooses among them using a DialogRPT-based reranker. NAISTeacher uses several different prompts, ranging from general to those targeting specific scenarios.

### 3.1 Zero-shot prompt

The base zero-shot prompt gives GPT-3.5-turbo the conversation history, along with scaffolding to encourage it to answer in a teacher-like way. The prompt reads as follows:

- (1) The following is a partial conversation between an English language learner and their teacher:

*(conversation)*

Can you give an example teacher follow-up to their previous message that would be helpful for the language learner? The message should be concise, and worded simply. It should either encourage the continuation of the current topic or gracefully transition to a new teacher-provided topic. Questions should be specific and not open-ended. Try to not sound like an assistant, but a teacher, in charge of the flow of the lesson.

The prompt went through several iterations to address problems such as answering questions too directly and sounding too much like an assistant. It started as a simple one-sentence prompt and we manually tested additions one-by-one on a subset of 3–5 conversations to address a number of challenges observed in the responses:

- the response being too thorough or technical;
- the end of the response not engaging the learner;
- questions being too open-ended, i.e. Is there anything else you're unsure about or need help with?

Our final prompt requests responses that are concise, encourage student engagement, and sound like a teacher rather than an assistant. We found that responses to this prompt tended to be a manageable length and level of detail, and they invite the student to engage with the conversation further and think about the topic in more depth.

See [Appendix A](#) for details on the iterations we tested for the zero-shot prompt.

### 3.2 Few-shot prompts

For the few-shot prompts, we provide one example teacher response that emulates the type of response we want, and one example of a response we would like to avoid. We used three different methods for choosing the positive and negative examples: handcrafted, generative, and iterative. See [Table 1](#) for the results of the three different methods.

#### 3.2.1 Handcrafted examples

We experimented with short manually written examples such as the following:

- (2) *Concatenation of prompt (1) and the following:*

Good example: 'Can you make a sentence using 'within'?' Bad example: 'Do you have any questions about prepositions?'

The goal of these examples was to get the system to ask questions to maximize learning for the student, but to not allow those questions to get so general that the student is most likely to respond with a short, one-word answer.

#### 3.2.2 Generated examples

Inspired by chain-of-thought prompting ([Wei et al., 2022](#)), we asked GPT-3.5-turbo to first consider what makes a good teacher response, and then integrated the answer back into the prompt.

For the case of teacher reply, we used GPT-3.5-turbo to generate a prompt with a 1-shot example pair, one positive, one negative. First GPT-3.5-turbo was asked:

- (3) How does a teacher sound when responding to a student? What kinds of things would teachers say that chatbots would not? What do they not say? In your response provide an example of a response that sounds like a teacher and one that sounds like a chatbot? Respond succinctly.

	BERTScore	DialogRPT
(1) Zero-shot, $k = 1$	70.91	36.51
(1) Zero-shot, $k = 3$	70.04	36.27
(2) Handcrafted 1-shot, $k = 3$	70.31	35.46
(4) Generated 1-shot, $k = 1$	70.88	38.26
(5) Iterative 1-shot, $k = 1$	<b>71.55</b>	<b>40.94</b>
Reference	1.0	32.81

Table 1: The results on the full test set of candidates generated with several different prompts.  $k$  is the number of candidates solicited by the prompt. For  $k > 1$  the score is the average of all the candidates with no reranking.

The answer to this question was integrated into the zero-shot prompt. Here is the final prompt with the GPT generated portion in bold.

- (4) The following is a partial conversation between an English language learner and their teacher:

*(conversation)*

They are in the middle of a lesson. Can you give a possible way the teacher could respond?

Remember: **A teacher typically sounds knowledgeable, authoritative, and focused on guiding and instructing students. They may use formal language and provide detailed explanations. Teachers often offer constructive feedback, encourage critical thinking, and ask probing questions to stimulate learning.**

**Example of a teacher-like response:** "That's a great observation, but let's delve deeper into the topic. Can you provide some evidence to support your claim?"

**A chatbot, on the other hand, may sound more informal and conversational. It tends to provide general information or brief responses without much elaboration.**

**Example of a chatbot-like response:** "Interesting! Tell me more."

**Teachers typically avoid expressing personal opinions or biases. They also refrain from engaging in casual banter or unrelated conversations to maintain a professional and educational atmosphere.**

#### 3.2.3 Iterative examples

The third method of generating examples for the prompt was an iterative approach. For this, we first used the zero-shot prompt (1) to generate responses for all conversations in the data. Next we used DialogRPT to score the responses and selected the highest and lowest scoring responses as the positive and negative examples respectively. In the final prompt we do not provide the full conversation context that led to the example, rather we use just the positive and negative response examples

themselves appended to the end of the zero-shot prompt. The final prompt is as follows:

(5) *Concatenation of prompt (1) and the following:*

Here is an example of an exceptional teacher follow-up:

"Great job, student! Just a small correction, we should use the present tense verb "built" instead of "build" since the construction has already been completed. So the correct sentence is: "The International Space Station is built by NASA." Keep up the good work! Now, let's move on to a new topic - let's talk about your favorite hobbies. Can you tell me what activities you enjoy doing in your free time?"

Here is an example of a poor teacher follow-up:

"That's an interesting observation about poshness. Can you think of any examples of British accents that might be associated with poshness?"

The idea behind this was to optimize for high scoring prompts on DialogRPT, and the results show an improvement in average DialogRPT scores. See Table 1 for the automatic evaluation of the responses from each of these prompts.

### 3.3 Prompting for multiple candidates

By modifying `Can you give an example ( $k = 1$ )` to `Can you give three examples ( $k = 3$ )`, we were able to illicit three replies at once. While originally implemented in an attempt to save both time and money when generating multiple candidates, this technique had multiple unexpected positive effects: it produced shorter responses in line with the length of the reference sentences, increased the diversity of the output as compared to running the same prompt twice, and allowed us to filter out candidates with profanity or references to things that would be inappropriate in a text chat (i.e. pronunciation practice) in a post-processing step (see 3.5 Post-processing).

The real teachers responded with comparatively short responses: the mean response length of the references was 23 words. Without specifying length requirements, `GPT-3.5-turbo` would return longer, more thorough responses, averaging over 35 words. By requesting three responses at once, the options shortened naturally to just over 23 words in the zero-shot case. While shorter responses may not work as well for other tasks, on this task, the shorter responses more closely matched the length of the reference sentences. See Table 2 for a comparison of sentence length and automatic evaluation.

In addition to making the responses more concise, requesting multiple candidate responses at

	avg. characters	avg. words
(1) Zero-shot, $k = 1$	205.16	35.55
(1) Zero-shot, $k = 3$	137.09	23.67
Reference	126.26	23.01

Table 2: The average length of the responses of the zero-shot prompt with  $k = 1$  and  $k = 3$  compared to the reference teacher responses.

	distinct-1	distinct-2
(1) Zero-shot, $k = 1$	0.62	0.84
(1) Zero-shot, $k = 3$	0.77	0.91

Table 3: The average percentage of distinct unigrams (distinct-1) and bigrams (distinct-2) present in the candidate sentences.

once also introduced more diversity to the responses compared to requesting a single response three times. There are many possible ways for a teacher to respond in a given situation, and generating many candidate responses allows the system to choose the best one. However, there is not much value to be gained from choosing between very similar candidates. When the candidates are more diverse, there is more chance of generating a really high quality response. We found that the responses generated by the  $k = 3$  prompts were more diverse compared to multiple inferences using the same prompt.

To measure diversity, we calculated the distinct unigrams and bigrams present in the candidates, normalized by dividing by the total number of words in the candidates following Li et al. (2015). The results shown in Table 3 demonstrate that there is very little overlap between the candidates generated by the  $k = 3$  prompts.

While using  $k = 3$  prompts generates shorter and more diverse candidate, it does come at the cost of a slight performance hit on the automated metrics. For further investigation into possible causes of this see 5.2 DialogRPT Length Bias Investigation. We used a combination of  $k = 1$  and  $k = 3$  prompts to balance the output of the full system.

### 3.4 Adaptations for teacher replies vs. continuations

Upon manual evaluation of the output, we found that some prompts, including the iterative 1-shot prompts (5), are better suited to generating teacher replies (the teacher turn following a student turn) than teacher continuations (the teacher turn following a teacher turn). We split the task of responding

	BERTScore	DialogRPT
(1) Zero-shot, $k = 1$	71.34	37.25
(1) Zero-shot, $k = 3$	70.32	36.21
(2) Handcrafted 1-shot, $k = 3$	70.62	35.53
(4) Generated 1-shot, $k = 1$	71.52	39.25
(4) Generated 1-shot, $k = 3$	70.98	37.39
(5) Iterative 1-shot, $k = 1$	<b>72.15</b>	<b>41.53</b>
(6) Reply, $k = 3$	70.82	34.67
(13) Reply long, $k = 3$	70.82	34.68
(7) Targeted transition, $k = 3$	70.32	37.21
Reference	1.00	32.31

Table 4: The results on the subset of the test data where the last speaker was the student (teacher reply).

to the student from the task of generating teacher follow-up utterances and evaluated the general systems separately on this subset of the data. At inference time, the system selects which prompts will be used for the ensemble based on the speaker in the final turn of the provided conversation history; if the role was a student, it chooses the prompts for teacher reply, and if the role was a teacher, it chooses the prompts for teacher continuation.

### 3.4.1 Teacher reply

The case of teacher reply can be thought of as the default case. The prompts described thus far generated reasonable responses for this case. Table 4 contains the results of the different prompts used for teacher reply evaluated only on the subset of the test data where the generated teacher utterance is responding directly to the student. The iterative 1-shot (5) prompt scored highest on both automated metrics, but many of the prompts generated several high quality candidates that were chosen for the final output of the system.

Two additional zero-shot prompts were engineered specifically to target the case of teacher reply, and were only used for conversations where the final utterance was from the student:

- (6) Here is a partial conversation between a student and their teacher during a private English lesson:

*(conversation)*

Can you give three possible ways the teacher could respond to continue the lesson? Use Simple English. While the conversation might be about culture or other topics, the point is to practice English. Each teacher response should:

1. Acknowledge what the student said, and demonstrate understanding.
2. Be helpful to the student, without answering directly. Give hints to help the student think for themselves.
3. Encourage the student to respond with an exercise or question.

Respond without preamble, just number them.

This prompt was engineered to target two of the three criteria for human evaluation from Tack and Piech (2022): Does it sound like it understands the student? Is it helpful for the student?

The second prompt was engineered to allow for changes of topic in the conversation. Originally this prompt was applied to the full dataset. However, we found that the responses were less effective in the case of teacher continuation, and so in the end they were used only for the case of teacher reply.

- (7) The following is a partial live chat between a teacher and a student learning English. They are in the middle of a lesson. Can you provide 3 possible ways the teacher could wrap up the current conversation and start an exercise or new topic of discussion?

Remember: Teachers often use specific language and techniques that chat bots have difficulty replicating. For example, they may ask open-ended questions to encourage critical thinking and engagement, provide specific feedback on a student's work, or offer personalized guidance based on a student's strengths and weaknesses. Teachers have a specific agenda for each lesson, such as practicing a specific grammar point or vocabulary. Try to understand what's happening in the conversation and what the teacher's goal is for the lesson.

If the goal is unclear, you can assume that the teacher wants to move on to a new topic or exercise.

Don't ask questions that are overly general such as "Is there anything else you'd like to talk about?"

Here is the conversation so far:

*(conversation)*

### 3.4.2 Teacher continuations

The task of generating teacher continuations proved more challenging than generating teacher replies. As dialogue systems are not typically trained on this task, it is particularly prone to producing spurious student responses between turns, i.e. responding as the student before providing a teacher response. One of the most common problems that arose in this subtask was that the generated candidate continuations would try to respond to the previous utterance as if it were a different speaker. To address this, several zero-shot prompts were engineered to cover possible reasons for a teacher to send a follow-up message before the student takes a turn. These prompts were carefully crafted during the development phase to ensure there were fewer spurious student responses.

The results of the generally applied prompts were manually evaluated to judge the appropri-



	BERTScore	DialogRPT
(1) Zero-shot, $k = 1$	69.89	34.82
(1) Zero-shot, $k = 3$	69.47	36.38
(2) Handcrafted 1-shot, $k = 3$	69.69	35.31
(8) Continue, $k = 3$	70.09	36.29
(14) Continue long, $k = 3$	69.49	34.76
(10) Generated 1-shot, $k = 3$	69.01	32.82
(11) Exercise, $k = 1$	<b>70.20</b>	<b>36.42</b>
(12) Conversation, $k = 1$	69.37	32.11
Reference	1.00	33.81

Table 5: The results on the subset of the test data where the last speaker was the teacher (teacher continuation).

ateness of the output for this subtask. Despite a high performance on the automated metrics, we removed the iterative 1-shot prompt (5) responses from consideration in the teacher continuation case. This decision was made because the candidates frequently sounded as if the student said something between the teacher utterances. The results of evaluation on the teacher continuation subset of the test data are shown in Table 5<sup>1</sup>.

The first prompt generated specifically for use in teacher continuations was a simple one.

- (8) Here is a partial conversation between a student and their teacher during a private English lesson:

*(conversation)*

Can you give three possible ways the teacher could continue their response? Use simple English.

Similar to the prompt generated for the teacher reply (4), we used GPT-3.5-turbo to generate a detailed prompt for the case of teacher continuation. First GPT-3.5-turbo was asked:

- (9) In the following conversation, the teacher has already sent a message. As this is a live chat, they want to send another message right away, before the student has a chance to reply. What might be some reasons why they want to follow-up on their previous message?

The answer was embedded in a new zero-shot prompt:

- (10) The following is a partial live chat between a teacher and a student learning English. They are in the middle of a lesson, and the teacher has already sent a message, but wants to follow-up. **There could be various reasons why the teacher wants to follow-up on their previous message before the student has a chance to reply. Here are some possibilities:**

- The teacher may have realized that their previous message contained some inaccuracies or omissions, and they want to correct or clarify their**

<sup>1</sup>While we report them here, we consider the DialogRPT scores to be unreliable as the models were not trained to evaluate this subtask.

**statement to avoid confusion.**

**2. The teacher may have received new information or thought of a better way to explain something, and they want to add to their previous message to provide a more complete answer.**

**3. The teacher may want to check if the student has any further questions or needs more explanation on the topic, and they want to encourage further discussion by sending a follow-up message.**

**Regardless of the reason, the teacher’s follow-up message can help ensure that the student fully understands the topic being discussed and feels comfortable asking questions and engaging in the conversation.**

Can you provide 3 possible follow-up messages the teacher could write?

Use simple English. The response should sound like a teacher, not an assistant. Good example: 'Can you make a sentence using 'within'?' Bad example: 'Do you have any questions about prepositions?'. The response should be helpful for the student and show that the teacher understood the student.

Here is the conversation so far:

*(conversation)*

### 3.4.3 Specific teacher continuation scenarios

Two prompts were designed for teacher continuation. The first prompt is used when the teacher has not provided an exercise or question for the student to respond to. In a lesson, it is typically the teacher’s responsibility to keep the student engaged, the conversation flowing, and the lesson on track. With this in mind, we asked GPT-3.5-turbo to check if the teacher has already asked a question, and if not, to provide one.

- (11) Here is a partial conversation between an English student and their teacher:

*(conversation)*

In the last utterance, did the teacher ask a question? If not, please provide one that would be appropriate. If they were in the middle of an exercise, what should they say to continue the exercise? The question or prompt should be simple. Don’t be too verbose or open ended. Good example: "What else is 'surprising'?" Bad example: "Is there anything else you’d like to know?"

Respond in the following format:  
Teacher asked a question: (yes/no)  
Question or prompt:

Similarly, a prompt was generated for the case that the teacher and student were engaged in more casual chitchat rather than exercises.

- (12) Here is a partial conversation between an English student and their teacher:

(conversation)

Were they in the middle of a conversation? If so, what should the teacher say to continue the conversation? The question or prompt should be simple and not use terminology such as 'collocations'. Don't be too verbose or open ended. Good example: "What else is 'surprising'?" Bad example: "Is there anything else you'd like to know?"

Respond in the following format:  
Conversation: yes or no  
Teacher:

Both of the above prompts required a bit more post-processing, but explicitly requesting a format in the prompt simplified this task.

### 3.5 Post-processing

The raw outputs of GPT-3.5-turbo contained inconsistent formatting, sometimes including quotes around the sample response, or sometimes prefixing `Teacher:` to the reply. The prompts that asked for multiple responses resulted in a numbered list, sometimes formatted `1:`, `2:`, and sometimes `1)`, `2)`. Occasionally this would include preamble such as `The teacher could reply:`. Prompts (11) and (12) both specified a pattern of output, and required slightly different post-processing to extract the relevant information and text. In the case of (11), if an exercise was already provided then there was no need to save the suggested candidate.

Post-processing was done on all of the GPT-3.5-turbo outputs to make the format more consistent and to separate the replies when multiple were requested. Separating the replies on the  $k = 3$  prompts was as simple as splitting on the new line, discarding lines that did not start with a number, and removing the numbers with the regular expression `/^\d+[\.\.]\s+/. If the remaining text was enclosed in quotes, the quotes were removed. If the remaining text started with teacher:, the prefix was removed. If the string started with student:, the entire candidate response was flagged as a student utterance and removed. If the response included any of the following phrases that indicate a request for a verbal response, we removed it from the list of candidates:`

```
try repeating
repeat after me
practice pronunciation
```

	contributions to final
(1) Zero-shot, $k = 1$	20
(2) Handcrafted 1-shot, $k = 3$	35
(4) Generated 1-shot, $k = 1$	22
(5) Iterative 1-shot, $k = 1$	72
(6) Targeted reply, $k = 3$	12
(7) Targeted transition, $k = 3$	40
(8) Continue, $k = 3$	17
(14) Continue long, $k = 3$	14
(10) Generated 1-shot, $k = 3$	27
(11) Exercise, $k = 1$	11
(12) Conversation, $k = 1$	3
Total	273

Table 6: The number of responses from each prompt that were chosen by the reranker for the final output.

For the final system, which chooses between utterances generated by several different prompts, each candidate response was run through a profanity filter<sup>2</sup> and discarded in the case of a profanity being detected.

### 3.6 Reranking

We used a very simple reranker that chose the candidate response with the highest DialogRPT score. The final score was calculated as a composite of subscores.

$$D_{\text{final}} = (D_{\text{updown}} + 0.48D_{\text{depth}} - 0.5D_{\text{width}}) \times (0.5D_{\text{vs-random}} + 0.5D_{\text{vs-machine}}) \quad (1)$$

Each of the scores was calculated with a different HuggingFace model<sup>3</sup>:

- microsoft/DialogRPT-updown ( $D_{\text{updown}}$ ),
- microsoft/DialogRPT-depth ( $D_{\text{depth}}$ ),
- microsoft/DialogRPT-width ( $D_{\text{width}}$ ),
- microsoft/DialogRPT-human-vs-rand ( $D_{\text{vs-random}}$ ),
- microsoft/DialogRPT-human-vs-machine ( $D_{\text{vs-machine}}$ ).

Table 6 contains the number of responses that came from each of the prompts in the submitted answers to the test set.

## 4 Results

Despite never using the reference sentences for training or fine-tuning, our system received the highest BERTScore and second highest DialogRPT score on the evaluation data, giving us the highest average rank in the automated metrics.

<sup>2</sup><https://github.com/rominf/profanity-filter>

<sup>3</sup>All DialogRPT models can be found here: <https://github.com/golsun/DialogRPT>

	teacher-like	understanding	helpful
NAISTeacher	<b>2.16</b>	<b>2.07</b>	<b>1.87</b>
Reference	3.11	3.10	3.09

Table 7: The average ranking results from the human evaluation. The best possible score is 1 and worst is 4.

The human evaluation of the top three teams was carried out on the Prolific crowdsourcing platform, where our system was compared against the other two systems as well as against the reference teacher utterances. The raters chose the best response on three criteria: (1) which was more likely said by a teacher, (2) which shows a better understanding of the student, and (3) which helps the student more. Table 7 contains the comparison of our system and the reference.

Our system scored the highest out of the three manually evaluated systems on all three criteria, and especially highly on the helpfulness criteria.

## 5 Discussion

### 5.1 Alternatives explored

We experimented with several different approaches, including prompt engineering with GPT-3.5-turbo (Brown et al., 2020), zero-shot BlenderBot (Roller et al., 2021), and fine-tuning GPT-2 (Radford et al., 2019). We found that while fine-tuning GPT-2 led to higher BERTScores, prompting GPT-3.5-turbo led to higher DialogRPT scores and sounded much more like a teacher upon manual inspection. Tack and Piech (2022) found that without fine-tuning or prompt tuning, BlenderBot was more successful than GPT-3 on this task. However, we discovered that with more specific prompting, the output of GPT-3.5-turbo could be influenced, so our final system only uses this.

### 5.2 DialogRPT Length Bias Investigation

One of the drawbacks of using  $k = 3$  prompts was that they scored lower on the automated metrics than their  $k = 1$  counterparts on average. We inspected the output manually, and on average preferred the responses from the  $k = 3$ .

To investigate if the lower performance of  $k = 3$  prompts was due to a bias in DialogRPT towards longer answers, we added a requested length to two targeted zero-shot prompts: (6) and (8).

(13) *Concatenation of prompt (6) and the following:*

Each response can be 3-5 sentences.

Teacher	number 1 I mean! I;m just giving you a little bit of help by telling you the first word you need from the list ('Australia')
Student	ok, could I use its used a half Australia of electricity produced coal?
Teacher	OK so: 'Australia used...(you continue JUST with the words I've given you)...'
Teacher	Australia used approximately half of the electricity it produced from coal. Keep going!

Figure 3: An example of a teacher continuation that gives the answer instead of scaffolding for the student.

(14) *Concatenation of prompt (8) and the following:*

Each response can be 3-5 sentences.

By adding the length specification of 3-5 sentences, the responses generated by the model become longer, but scored no higher on DialogRPT, leading us to conclude there is no evidence of bias towards longer answers. See Table 8 for detailed results.

### 5.3 Common errors

An illustration of a common remaining error can be seen in Figure 3. The student has yet to attempt to complete the exercise, and instead of scaffolding the student's attempt to complete it, the teacher responds with the answer.

This could be fixed with a better reranking algorithm, as DialogRPT often scores the responses that contain the correct answer higher than those that scaffold the learner.

For example, the chosen response scored 68.17, while an alternative response that scaffolds better scored 50.34: Let's focus on the structure of the sentence next. Remember to use the correct verb form after "Australia used". Also, instead of "half Australia", we would say "half of Australia's". Could you try revising your sentence to reflect these changes?

## 6 Conclusion

The reranker we built was very simple. It selected the highest-scoring response according to the automated metric, DialogRPT. However, preliminary manual evaluation did not always align with DialogRPT. The metric often prefers complete answers that do not encourage student engagement over responses that aim to help the student answer the question for themselves.

	avg. characters	avg. words	BERTScore	DialogRPT
(1) Zero-shot, $k = 1$	205.16	35.55	70.91	36.51
(1) Zero-shot, $k = 3$	137.09	23.67	70.04	36.27
(6) Reply, $k = 3$	170.76	29.67	83.29	36.39
(13) Reply long, $k = 3$	231.58	40.77	70.82	34.68
(8) Continue, $k = 3$	145.43	25.52	70.09	36.29
(14) Continue long, $k = 3$	238.71	42.26	69.49	34.76
Reference	126.26	23.01	1.00	32.81

Table 8: The average length of the responses of prompts with  $k = 1$  and  $k = 3$ , as well as those with a length of 3-5 sentences specified in the prompt.

With additional time, we would like to develop a model capable of classifying the extent to which a generated response reflects a teacher’s style. This model could take into account whether the response effectively balances helpfulness with scaffolding independent thought, as well as the degree to which it demonstrates an understanding of the student’s needs. Such a model would lead to improved performance of the reranker.

The AI teacher response generator we created still needs improvement before it can become a fully functional teacher chatbot. The responses it generates can be excessively detailed at times due to the automated metrics used, which prioritize comprehensive responses. When integrated into an assistant, it may seem as if the responses are repetitive or that there is no well-designed lesson plan in place.

As we were building the system, we kept in mind how it was to be evaluated. That is, by machine first checking for similarity to a reference answer as well as usefulness and relevance to the conversation, then by humans evaluating how teacher like, understanding, and helpful the response was. We used a combination of  $k = 1$  prompts which scored higher on the automated metrics, and  $k = 3$  prompts which produced shorter responses that we preferred on manual evaluation.

While the automated metrics taken together align with human evaluators’ judgments, DialogRPT alone does not always correspond with human judgment. The DialogRPT score models were trained on Reddit, which follows a different format than live chat. Reddit is an asynchronous format, meaning that it tends to have longer, more complete responses. On the other hand, synchronous chat-based lessons feature multiple consecutive turns, as it is more common in instant messaging to break up longer thoughts into smaller turns. DialogRPT was not trained to judge the continuation of a response, which made it less reliable as a reranker for teacher

continuations, in particular.

When chatting, it’s not necessary to always respond with a detailed message. The reference teacher responses offer a mix of quick replies, corrections, elaborations, practice activities, and clarifications, among others. In the future, we would like to incorporate more of the conversational moves that real teachers use in these types of exchanges.

In conclusion, our approach to generating teacher utterances in an educational dialogue for the BEA2023 shared task used a pretrained language model and an ensemble of prompts to generate multiple candidates, which we then reranked using automated metrics. We experimented with different techniques for generating few-shot prompts and found that an iterative approach was the most effective. Our system achieved the highest averaged ranked scores in both the automated and human evaluation rounds. Overall, our approach shows promise for generating effective and helpful teacher utterances in educational dialogues.

## Limitations

A limitation of our approach is that it relies heavily on the quality and relevance of the prompts used. The prompts were engineered based on observations made in the training data and this approach may not work if the prompts are not representative of the corpus. Finally, our approach may not be suitable for all types of teacher-student dialogues and may require modifications for different contexts or domains.

One possible concern with the techniques mentioned in this paper is the limited reproducibility of OpenAI’s language models, such as GPT-3.5-turbo. The weights of these models are proprietary and not publicly accessible, which makes it challenging to replicate the findings of earlier research or expand on them.

## Ethical Considerations

AI-generated teacher utterances may contain bias, which may become apparent particularly in exercises or chit-chat.

In this project, we took steps to avoid profanity in the AI-generated responses, but similar protection against bias should be put into place. Additionally, human evaluators should be used to assess the quality of the AI-generated responses and to identify any potential biases.

We recognize that language models like GPT-3.5-turbo are trained on large datasets that reflect the biases and prejudices present in society. As there is always a risk of perpetuating these biases when using generative AI for dialogue systems it is important to evaluate the AI-generated responses for potential biases and to take steps to correct them.

## Acknowledgements

We are very grateful for the thoughtful and helpful feedback from our reviewers, and to Xincan Feng, Iqra Ali, and Hidetaka Kamigato for their support and feedback.

## References

- Dennis Aumiller and Michael Gertz. 2022. [UniHD at TSAR-2022 shared task: Is compute all we need for lexical simplification?](#) In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 251–258, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.
- 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.
- Andrew Caines, Helen Yannakoudakis, Helen Allen, Pascual Pérez-Paredes, Bill Byrne, and Paula Buttery. 2022. [The teacher-student chatroom corpus version 2: more lessons, new annotation, automatic detection of sequence shifts](#). In *Proceedings of the 11th Workshop on NLP for Computer Assisted Language Learning*, pages 23–35, Louvain-la-Neuve, Belgium. LiU Electronic Press.
- Andrew Caines, Helen Yannakoudakis, Helena Edmondson, Helen Allen, Pascual Pérez-Paredes, Bill Byrne, and Paula Buttery. 2020. [The teacher-student chatroom corpus](#). In *Proceedings of the 9th Workshop on NLP for Computer Assisted Language Learning*, pages 10–20, Gothenburg, Sweden. LiU Electronic Press.
- Xiang Gao, Yizhe Zhang, Michel Galley, Chris Brockett, and Bill Dolan. 2020. [Dialogue response ranking training with large-scale human feedback data](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 386–395, Online. Association for Computational Linguistics.
- Ann Lee, Michael Auli, and Marc’Aurelio Ranzato. 2021. [Discriminative reranking for neural machine translation](#). 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 7250–7264, Online. Association for Computational Linguistics.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. [A diversity-promoting objective function for neural conversation models](#). *CoRR*, abs/1510.03055.
- Franz Josef Och and Hermann Ney. 2002. [Discriminative training and maximum entropy models for statistical machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 295–302, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. [Recipes for building an open-domain chatbot](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.
- Libin Shen, Anoop Sarkar, and Franz Josef Och. 2004. [Discriminative reranking for machine translation](#). In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 177–184, Boston, Massachusetts, USA. Association for Computational Linguistics.
- Anais Tack, Ekaterina Kochmar, Zheng Yuan, Serge Bibauw, and Chris Piech. 2023. The BEA 2023 Shared Task on Generating AI Teacher Responses in Educational Dialogues. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building*

*Educational Applications*, page to appear, Toronto, Canada. Association for Computational Linguistics.

Anais Tack and Chris Piech. 2022. The AI Teacher Test: Measuring the Pedagogical Ability of Blender and GPT-3 in Educational Dialogues. In *The 15th International Conference on Educational Data Mining*, page accepted.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. [Chain of thought prompting elicits reasoning in large language models](#). *CoRR*, abs/2201.11903.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. [Bertscore: Evaluating text generation with BERT](#). *CoRR*, abs/1904.09675.

## A Zero-shot iterations

The zero-shot prompt went through several iterations. We present six versions along with the responses for the following example conversation:

Teacher: keywords they focus on in that page i mean  
Student: OK, I'll do it now  
Teacher: ok  
Student: ...provocative, and supplied +startling+ messages about psychology...  
Student: Is it enough this piece of text to work on it? the word I don't know is startling  
Teacher: ok thanks....'startling' does it mean: obvious or surprising do you think? is there a context in the text to help you?

- (15) Can you give an example teacher response that would be helpful for the language learner and encourage the continuation of the conversation?

NAISTeacher: Great job providing those keywords! Let's take a closer look at the word "startling." It means something that is very surprising or shocking. For example, "The news of the unexpected victory was startling to everyone." In the context of the text you provided, it seems that the messages about psychology were surprising and caught your attention. Keep up the good work! Is there anything else you're unsure about or need help with?

The response from prompt (15) on this example reveals several potential problems:

- It answered the teacher's question before allowing the student to reply.
- It helpfully provides an example sentence, but the inclusion of this makes the reply unnaturally long.
- It sounds as though the teacher is unfamiliar with the text they assigned.
- The final question sounds like an assistant or a chatbot, not a teacher. While it is helpful and pleasant, it lacks a sense of agency or authority in the situation.

With these problems in mind, we iterated on the prompt.

- (16) Can you give an example teacher **follow-up to their previous message** that would be helpful for the language learner and ~~encourage the continuation of the conversation?~~

NAISTeacher: Good job identifying those keywords! 'Startling' means surprising or shocking. In this context, it suggests that the messages about psychology were not only provocative, but also had an element of surprise or shock to them. Keep up the good work!

With the exclusion of and encourage the continuation of the conversation the response ends too finally. We want a way to keep the student engaged and active in the conversation.

- (17) Can you give an example teacher follow-up to their previous message that would be helpful for the language learner? **The message should encourage the continuation of the current topic, or gracefully transition to a new topic.**

NAISTeacher: Great job using the keywords from the text, it seems like you have a good understanding of the content. To answer your question, 'startling' means surprising or unexpected. In the context of the text, it suggests that the messages about psychology were particularly surprising or unexpected. Keep up the good work! Is there anything else you'd like to discuss related to this text or another topic?

The end of the response improved, but is too open ended, and sounds unnatural for a teacher.

- (18) Can you give an example teacher follow-up to their previous message that would be helpful for the language learner? The message should encourage the continuation of the current topic, or gracefully transition to a new teacher-provided topic. **Try to not sound like an assistant, but a teacher, in charge of the flow of the lesson.**

NAISTeacher: Great job identifying those keywords! To answer your question, 'startling' means surprising or shocking. It's often used to describe something that catches you off guard or is unexpected. In this context, it seems like the article is trying to grab the reader's attention with provocative and surprising messages about psychology.

Now that we've clarified the meaning of 'startling,' let's continue with our analysis of the text. Can you tell me more about what you think the article is trying to convey? Do you have any questions or ideas about the content?

The question at the end is much more specific and teacher-like, but there remains the problem of the teacher over-explaining and answering a question too directly. We wanted the teacher to encourage the student to think rather than just read an explanation.

- (19) Can you give an example teacher follow-up to their previous message that would be helpful for the language learner? **Instead of answering the student's question directly, give hints and encourage the student to think for themselves.** The message should encourage the continuation of the current topic, or gracefully transition to a new teacher-provided topic. Try to not sound like an assistant, but a teacher, in charge of the flow of the lesson.

NAISTeacher: Great job, you're making progress! You're right that "startling" is not a very common word, but it's definitely a useful one to know. To answer your question, "startling" means surprising, or even shocking. So when you see that word in a text, it usually means that something unexpected or surprising is being described. In this context, it seems that the page you were reading contained messages about psychology that were not only

provocative, but also surprising or shocking in some way.

It's always a good idea to try to understand the context of a word in order to better understand its meaning. Sometimes, the words that come before or after it can help you understand what it means. For example, in this case, the word "provocative" suggests that the messages were intended to be controversial or challenging, and "startling" adds to this by suggesting that the messages were even more unexpected than one might expect from a provocative message.

Keep up the good work, and don't hesitate to ask if you have any more questions or if you want to practice using these words in conversation or writing!

The response to this prompt is way too long, and the inclusion of *Instead of answering the student's question directly, give hints and encourage the student to think for themselves* did not solve the problem. We remove that and add the stipulation that the response be concise.

- (20) Can you give an example teacher follow-up to their previous message that would be helpful for the language learner? **The message should be concise, and worded simply.** ~~Instead of answering the student's question directly, give hints and encourage the student to think for themselves.~~ It should either encourage the continuation of the current topic or gracefully transition to a new teacher-provided topic. **Questions should be specific and not open-ended.** Try to not sound like an assistant, but a teacher, in charge of the flow of the lesson.

Great job using the keywords from the text! "Startling" means surprising. In this context, it seems like the text contains provocative and surprising messages about psychology. Can you think of any other words that might be related to the topic of psychology?

We found this to be a reasonable response. It still has the problem of answering the question posed by the teacher, which we chose to address by splitting out teacher continuations (see [3.4.3 Specific teacher continuation scenarios](#)), but the response is a manageable length and invites the student to engage with the conversation further and think about the topic in more depth.



# The BEA 2023 Shared Task on Generating AI Teacher Responses in Educational Dialogues

Anaïs Tack

KU Leuven, imec

anaïs.tack@kuleuven.be

Ekaterina Kochmar

MBZUAI

ekaterina.kochmar@mbzuai.ac.ae

Zheng Yuan

King's College London

zheng.yuan@kcl.ac.uk

Serge Bibauw

Universidad Central del Ecuador

sbibauw@uce.edu.ec

Chris Piech

Stanford University

cpiech@stanford.edu

## Abstract

This paper describes the results of the first shared task on generation of teacher responses in educational dialogues. The goal of the task was to benchmark the ability of generative language models to act as AI teachers, replying to a student in a teacher–student dialogue. Eight teams participated in the competition hosted on CodaLab and experimented with a wide variety of state-of-the-art models, including Alpaca, Bloom, DialoGPT, DistilGPT-2, Flan-T5, GPT-2, GPT-3, GPT-4, LLaMA, OPT-2.7B, and T5-base. Their submissions were automatically scored using BERTScore and DialogRPT metrics, and the top three among them were further manually evaluated in terms of pedagogical ability based on Tack and Piech (2022). The NAISTeacher system, which ranked first in both automated and human evaluation, generated responses with GPT-3.5 Turbo using an ensemble of prompts and DialogRPT-based ranking of responses for given dialogue contexts. Despite promising achievements of the participating teams, the results also highlight the need for evaluation metrics better suited to educational contexts.

## 1 Introduction

Conversational AI offers promising opportunities for education. Chatbots can fulfill various roles – from intelligent tutors to service-oriented assistants – and pursue different objectives such as improving student skills and increasing instructional efficiency (Wollny et al., 2021). One of the most important roles for an educational chatbot is that of an AI teacher which helps a student improve their skills and provides more opportunities to practice. Recent studies suggest that chatbots have a significant effect on skill improvement, for example, in language learning (Bibauw et al., 2022). Moreover, the advances in Large Language Models (LLMs) open up new opportunities as such models have a potential to revolutionize education and significantly transform learning and teaching experience.

Despite these promising opportunities, the use of powerful generative models as a foundation for downstream tasks presents several crucial challenges, in particular, when such tasks may have real social impact. Specifically, in the educational domain, it is important to determine how solid that foundation is. Bommasani et al. (2021) (pp. 67–72) stresses that if we want to put such models into practice as AI teachers, it is of crucial importance to determine whether they can (a) speak to students like a teacher, (b) understand students, and (c) help students improve their understanding. Following these desiderata, Tack and Piech (2022) formulated the AI teacher test challenge: *How can we test whether state-of-the-art generative models are good AI teachers, capable of replying to a student in an educational dialogue?*

Building on the AI teacher test challenge, we have organized the first shared task on generation of teacher language in educational dialogues. The goal of this task is to explore the potential of NLP and AI methods in generating teacher responses in the context of real-world teacher–student interactions. Interaction samples were extracted from the *Teacher Student Chatroom Corpus* (Caines et al., 2020, 2022), with each training sample consisting of a dialogue context (i.e., several rounds of teacher–student utterances) and the teacher’s response. For each test sample, participants were asked to submit their best generated teacher response.

As the purpose of this task was to benchmark the ability of generative models to act as AI teachers, responding to a student in a teacher–student dialogue, submissions were first ranked according to popular BERTScore and DialogRPT metrics, and the top three submissions were then selected for further human evaluation. During this manual evaluation, the raters compared a pair of “teacher” responses along three dimensions: speaking like a teacher, understanding a student, and helping a student (Tack and Piech, 2022).

SPEAKER	UTTERANCE	
<b>Teacher:</b>	Yes, good! And to charge it up, you need to __ it ____	] DIALOGUE CONTEXT
<b>Student:</b>	...	
<b>Teacher:</b>	connect to the source of electricity	
<b>Student:</b>	i understand	
<b>Teacher:</b>	plug it __?	
<b>Student:</b>	in	= REFERENCE RESPONSE
<b>Teacher:</b>	yes, good. And when the battery is full, you need to ____ (disconnect it)	

Figure 1: An example of a sample taken from the *Teacher-Student Chatroom Corpus*

## 2 Materials and Methods

The shared task used data from the *Teacher-Student Chatroom Corpus* (TSCC) (Caines et al., 2020, 2022). This corpus comprises data from several chatrooms in which an English as a second language (ESL) teacher interacts with a student in order to work on a language learning exercise and assess the student’s English language proficiency.

### 2.1 Data Samples

Several samples were taken from each dialogue in the corpus. Each sample was composed of several sequential teacher-student turns (i.e., the preceding dialogue context) and ended with a teacher utterance (i.e., the reference response). Figure 1 shows an example of a sample taken from the corpus. As can be seen from this example, the samples were quite short, counting at most 100 tokens. Even though this restricted sample size inevitably posed an important limitation for training and testing, the length of each sample had to be capped at this specific limit in order to comply with the copyright license and terms of use of the corpus.

#### 2.1.1 Extraction

The samples were extracted with the following method. For each dialogue in the corpus, the sequence of utterances was iterated from the first to the last. If the speaker of an utterance at the current position was a teacher, the utterance was a potential reference response. In that case, a contextual window sequence was created for the reference candidate by recursively backtracking through the dialogue and adding the preceding utterances until the limit of 100 tokens was reached. Each utterance was tokenized with spaCy’s default tokenizer for English.<sup>1</sup> Once extracted, the sequence was added

<sup>1</sup><https://spacy.io/api/tokenizer>

to the set of samples for the dialogue on the condition that it had at least two utterances and more than one speaker. For example, if the teacher initiated the conversation, the algorithm would extract a window with only one speaker and no preceding utterances. Because this instance would not have been informative, it was ignored and not added to the set of data samples. A total of 7,047 data samples were extracted from the original dataset.

#### 2.1.2 Selection

Although the extracted data samples could have been randomly divided into training and test samples, such an approach would have been problematic. In fact, it would have been possible for a randomly selected test sample to contain a reference response otherwise observed in the dialog context of *another* randomly selected training or test sample (see Figure 2). A related issue was that the extraction algorithm produced samples that were also part of other samples, resulting in multiple nested or Russian doll-like ensembles (see Figure 3). Since a test set should never include references seen elsewhere in the data, special attention was paid to data splitting.

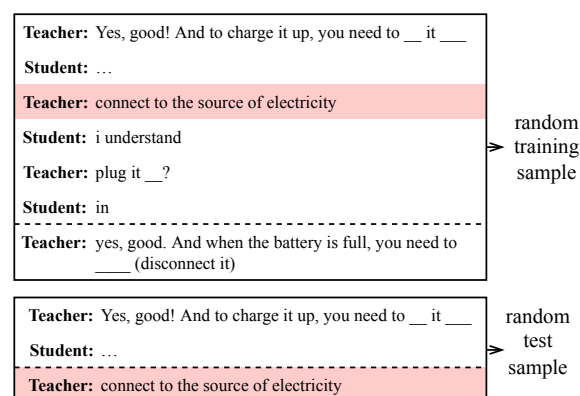


Figure 2: An example of a reference in a test sample observed in the context of a training sample

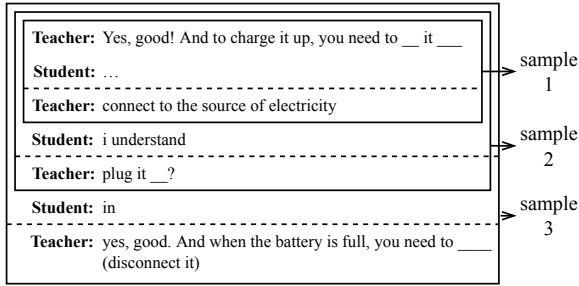


Figure 3: An example of a nested or Russian doll-like ensemble of data samples

The data samples were split into a training and test set with a more complex selection procedure. Three selection criteria were defined: (a) whether the reference response was labeled as *eliciting* and/or *scaffolding* ('yes'  $\Rightarrow$  better), (b) the number of distinct types of conversational organization (e.g., opening, closing, eliciting, scaffolding, and revision) that were added as labels to the reference response (more  $\Rightarrow$  better), and (c) the total number of tokens in the sample (more  $\Rightarrow$  better). The extracted data samples contained 1,400 nested ensembles (cf. Figure 3). The samples in each ensemble were sorted based on the three criteria above, and for each ensemble, only the best sample was selected. The remaining 4,864 samples were assigned to 2,457 training and 273 test slots with the *Hungarian algorithm* (Kuhn, 1955) based on the criteria above. Once the assignment was done, the training and test sets were verified for any potential conflicts (cf. Figure 2). Conflicts were resolved by using the criteria above to choose the best sample among the conflicting samples. Then, the assignment was run again on the remaining samples until no more conflicts could be detected. After the assignment was completed, the nested data samples that were discarded before were used to increase the size of the training set on the condition that they were not in conflict with the test set. Finally, the training set was randomly split into a 90% training and 10% held-out set. The number of samples included in the training and test sets are shown in Table 1.

Training set	3,052	
	90% training	2,747
	10% held-out	305
Test set	273	

Table 1: The number of training and test samples

## 2.2 Competition

The shared task was hosted as an [online competition](#) on the CodaLab platform (Pavao et al., 2022). Anyone participating in the shared task filled in a registration form, signed to comply with the terms and conditions of the shared task and the licensed TSCC data, and registered on the CodaLab platform. Participants could only be part of one team, while a team could have one or more participants.

### 2.2.1 Phases

The competition was run in two phases: a development and an evaluation phase. All deadlines were set to 23:59 Anywhere on Earth (UTC-12). Since CodaLab uses Coordinated Universal Time, all deadlines on the platform were adapted accordingly (i.e., set to the next day at 11:59 am UTC).

The development phase started on March 24, 2023, and ended on April 30, 2023. At the start of the development phase, participants received the training and held-out development data, which were available on the CodaLab platform. During the development phase, participants could submit their results for the held-out data and view their scores on the anonymized leaderboard. Sixty-three people filled in the registration form and registered on the CodaLab platform. Among them, 12 people actively participated in the development phase and submitted results on the held-out data. Three people submitted to the development phase after the evaluation phase had already started. In the end, 10 participants made at least one successful submission to the development phase. In total, 17 successful submissions were received ( $M_{\text{submissions}} = 1.7$  per participant). The leaderboard featured only the best successful submission per participant (see the metrics described below in Section 2.3.1).

The evaluation phase started on May 1st, 2023, and ended on May 5th, 2023. At the start of the evaluation phase, participants received the test data, which were available on the CodaLab platform. During the evaluation phase, participants could submit their results on the test data and view their scores on the anonymized leaderboard. In addition, six people filled in the registration form and registered on the CodaLab platform. Nineteen people actively participated in the evaluation phase and submitted their results on the test data. In the end, 10 participants from eight teams made at least one successful submission to the evaluation phase. In total, 19 successful submissions were received

( $M_{\text{submissions}} = 1.9$  per participant). Again, the leaderboard featured only the best successful submission per participant (see the metrics described in Section 2.3.1).

It should be noted that some people showed interest in the shared task but did not fully participate. Fifteen people filled in the registration form but did not request to join on the platform before the deadline, whereas 18 people requested to join on CodaLab but did not fill in the registration form. As a result, they could not be accepted into the competition because they did not sign to comply with the terms and conditions.

### 2.2.2 Teams and Systems

Eight teams made at least one successful submission to the final evaluation phase. The approaches taken by the teams were based on a range of state-of-the-art large language models (LLMs), including Alpaca (Team RETUYT-InCo), Bloom (RETUYT-InCo), DialoGPT (Cornell), DistilGPT-2 (DT), Flan-T5 (teams Cornell and TanTanLabs), GPT-2 (Cornell and Data Science-NLP-HSG), GPT-3 (NBU), GPT-3.5 Turbo (NAIST and aiitis), GPT-4 (Cornell), LLaMA (RETUYT-InCo), OPT-2.7B (RETUYT-InCo), and T5-base (Data Science-NLP-HSG). In addition, all teams experimented with zero- and few-shot learning, fine-tuning, and various prompting strategies. Several teams applied reinforcement learning (RL) (Cornell and Data Science-NLP-HSG), and some developed customized approaches to post-processing (NAIST) and data-driven prompt engineering (aiitis). All these approaches are summarized below and further detailed in the corresponding system papers.

**Team NAIST** Vasselli et al. (2023) participated in the shared task with the NAISTEACHER system, built on a pre-trained GPT-3.5 Turbo (Brown et al., 2020). They experimented with, on the one hand, zero-shot prompts and, on the other hand, few-shot prompts using either handcrafted, generative, or iterative examples of teacher responses. They also experimented with asking the model to generate either one response or several possible responses and compared the performance of their system in two settings: *teacher replies* (i.e., when the generated teacher utterance followed a student utterance) and *teacher continuations* (i.e., when the generated teacher utterance followed a teacher utterance). Finally, the candidate responses were post-processed (with a profanity filter and regular expressions) and

reranked with DialogRPT (see the shared task metrics in Section 2.3.1) in order to select the best response to be submitted for each test sample.

**Team NBU** Adigwe and Yuan (2023) participated in the shared task with the ADAIO system. They evaluated several GPT-3 models (Brown et al., 2020), designed various zero-shot and few-shot prompts to generate teacher responses, and also fine-tuned the models on the TSCC corpus. In addition, the team experimented extensively with various aspects of response generation by considering the roles of the participants, the teaching approaches taken by the tutor, and the specific teaching goals. The responses submitted to the competition were generated by a few-shot prompt-based method based on the *text-davinci-003* model.

**Team Cornell** Hicke et al. (2023) experimented with several generative models and various approaches, including few-shot in-context learning with GPT-4, fine-tuning of GPT-2 (Radford et al., 2019) and DialoGPT (Zhang et al., 2019), and fine-tuning of Flan-T5 (Chung et al., 2022) with RL (Ramamurthy et al., 2022) to optimize for pedagogical quality. Among these, GPT-4 achieved the best results on the shared task evaluation metrics (see Section 2.3.1). The team made two submissions to the leaderboard: one submission with responses generated by GPT-4, and another submission that included the same responses with a teacher prefix prepended to each of them ("teacher: <response>"). To distinguish between these submissions, the latter is referred to as GPT-4<sup>(TP)</sup> where TP stands for teacher prefix.

**Team aiitis** Omidvar and An (2023) introduced the Semantic In-Context Learning (S-ICL) model. Their aim was to address the challenges created by the use of out-of-the-box pre-trained LLMs, such as domain adaptivity and the high costs of fine-tuning. Their in-context learning approach consisted of providing an LLM (in this case, ChatGPT with the GPT-3.5 Turbo engine) with a prompt containing an instruction, a few labeled samples, and an unlabeled sample. The *semantic* component in the S-ICL model retrieved sufficiently similar samples from the training set, which were then integrated into the prompt fed to the LLM as labeled samples. The inclusion of relevant conversational samples in the prompt allowed the model to leverage available knowledge for generating teacher responses.

**Team RETUYT-InCo** Baladón et al. (2023) experimented with several open-source LLMs, including LLaMA (Touvron et al., 2023), Alpaca (Taori et al., 2023), OPT-2.7B (Gao et al., 2020a), and Bloom 3b (Scao et al., 2022). They explored fine-tuning techniques by applying the LoRA (Hu et al., 2021) method to the aforementioned LLMs. They tested several prompting strategies including few-shot and chain-of-thought approaches. Their method consisted of selecting the three most similar conversations from the training data using the  $k$ -nearest neighbors algorithm. These were then further integrated into the prompt for the few-shot learning scenario. The models submitted to the competition were trained using Alpaca LoRA with the few-shot approach, LLaMA 7B with engineered prompts fine-tuned with LoRA, and fine-tuned OPT-2.7B using preprocessing.

**Team Data Science-NLP-HSG** Huber et al. (2023) presented a simple approach of fine-tuning a language model with RL and utilized the novel NLPO algorithm (Ramamurthy et al., 2022) that masks out tokens during inference to direct the model towards generations that maximize a reward function. They used Hugging Face’s implementation of the T5-base model (Raffel et al., 2020) with 220 million parameters to generate the responses submitted to the competition.

**Team DT** This team experimented with fine-tuning the DistilGPT-2 model specifically for student–teacher dialogues. They divided the original training data using an 80/20 split and ran a three-epoch training process using the Adam optimizer along with a linear learning rate scheduler on the training subset. The remaining 20% were then used for rigorous evaluation using the shared task performance metrics. The team [released their model on Hugging Face](#) and plans to explore the potential of larger models like GPT-3 and GPT-4 in the educational dialogue domain in the future.<sup>2</sup>

**Team TanTanLabs** This team experimented with a zero-shot approach using Hugging Face’s Flan-T5 transformer model, a model instruction-finetuned on a mixture of tasks. Among the many prompting techniques tested, the one that worked best was the prompt used by the authors of the Flan-T5 model: “Read the dialog and predict the next turn.” For model inference, different decoding

techniques were tried (greedy, decoding by sampling with temperature, and beam search). Beam search was chosen because it was easy to control. Customized regular expressions were used to parse the model’s output. When the model didn’t produce any output, the filler word “Alright” was used. In the future, the team plans to experiment further with supervised fine-tuning using “chain of thought” reasoning instructions.<sup>3</sup>

## 2.3 Evaluation Procedure

The submissions made by the teams described above were evaluated in two stages. During the competition, all submissions were automatically scored with several dialogue evaluation metrics (see Yeh et al., 2021, for a comprehensive review). The teams used these metrics to optimize their systems before the end of the competition. After the competition ended, the final submissions were evaluated by human raters. Due to combinatorial constraints imposed by the human evaluation task (see Section 2.3.2), it was not possible for any number of submissions to be evaluated manually. For this reason, only the top three submissions on the automated metrics were targeted for human evaluation.

### 2.3.1 Evaluation Metrics

Yeh et al. (2021) reviewed several dialogue evaluation metrics that operate at the level of the individual turns (i.e., generated responses). However, many of these metrics required a complicated installation procedure. The following two metrics were used because they are well-known, could be easily installed, and their scores can be reproduced.

**BERTScore** (Zhang et al., 2020) was used as a metric for evaluating each generated response with respect to the reference (i.e., teacher) response. The metric matches words in submissions and reference responses by cosine similarity. BERTScore was computed with Hugging Face’s *evaluate* package and the *distilbert-base-uncased*<sup>4</sup> model. The resulting precision, recall, and F1 scores were averaged for all items in the test set.

**DialogRPT** (Gao et al., 2020b) was used as a reference-free metric for evaluating the generated response with respect to the preceding dialogue context. The metric consists of a set of ranked pre-trained transformer models proposed by Microsoft

<sup>2</sup>Written by Rabin Banjade and adapted by the authors

<sup>3</sup>Written by Tanay Gahlot and adapted by the authors

<sup>4</sup>The hashcode was *distilbert-base-uncased\_L5\_no-idf\_version=0.3.12(hug\_trans=4.28.1)*.

Research NLP Group. These metrics were aggregated for all items in the test set. The following dialog response ranking models were used:

**updown** likelihood that a response gets the most upvotes (mean of all items)

**human vs. rand** likelihood that a response is relevant for the given context (mean of all items)

**human vs. machine** likelihood that a response is human-written rather than machine-generated (mean of all test items)

**final** weighted ensemble score of all DialogRPT metrics (mean of all items)

Each submission was ranked from 1 (highest) to 10 (lowest) on each individual metric. The overall leaderboard rank was computed as the mean rank on BERTScore F1 and on DialogRPT final average. In case of a tie, the tiebreaker was the mean rank on the individual scores for BERTScore (precision, recall) and DialogRPT (updown, human vs. rand, human vs. machine).

### 2.3.2 Human Evaluation

The top  $k = 3$  submissions on the leaderboard were further evaluated by means of pairwise comparative judgments.<sup>5</sup> For each sample in the set of  $n = 273$  test items, the possible responses were combined in pairs such that the generated responses were either compared with the reference (i.e., teacher vs. AI) or between themselves (i.e., AI vs. AI). This resulted in  $\binom{k+1}{2} = 6$  pairs of responses for each test sample. Each pair was assessed by  $r = 3$  raters, which amounted to a total of  $\frac{(k+1)!}{2!(k+1-2)!}r = 4,914$  distinct assessments. These evaluations were collected via an online Qualtrics survey following a method described in Tack and Piech (2022) and further detailed below.

**Survey** In the introductory part of the survey, raters were given a short introduction, a consent form, and an example to familiarize themselves with the task at hand. In the central part of the survey, each rater was presented with a comparative

<sup>5</sup>In pairwise comparative judgments, multiple alternatives are evaluated by systematically assessing them in pairs. Each rater is presented with two alternatives at a time and makes a judgment about which one is better according to some criteria. These judgments are used to compute a relative ranking among the alternatives. This method has already been used for assessing dialogue systems (Li et al., 2019) and open-ended natural language generation (Pillutla et al., 2021).

judgment task of 20 items that were randomly and evenly selected from the set of  $n$  test samples. Each survey item included a pairwise comparison that was randomly and evenly selected from the  $\binom{k+1}{2}$  possible pairs for the chosen test sample. Each survey item had three components: the dialogue context, one comparison of two responses (A or B), and three questions targeting a pedagogic ability (*more likely said by a teacher, better understanding the student, and helping the student more*). For each question, the rater was asked to choose option A or B. The order in which the pairwise comparison was presented, was determined randomly so that any presentation order effects would be avoided.

**Raters** A sample of 298 raters were recruited from the Prolific crowdsourcing platform. The raters were screened based on several characteristics: (a) whether they were from a majority native English-speaking country,<sup>6</sup> (b) whether their native language was English, and (c) whether their employment sector was in education and training. The sample of raters was gender-balanced. Five raters were removed because the outlier detection described in Tack and Piech (2022) showed that they consistently picked the same option (A or B) for all questions throughout the survey.

**Ranking** For each item in the test set, the possible responses were ranked from 1 (highest) to 4 (lowest) for each of the three questions (*more likely said by a teacher, understanding the student better, and helping the student more*). The rank for each response (i.e., teacher or AI) was estimated with a Bayesian Bradley-Terry model and HMC-NUTS sampler as described in Tack and Piech (2022). Based on the set of draws produced by the HMC-NUTS sampler, the mean rank, standard deviation, and 95% highest density intervals (HDI) were computed for each item and for each response.

## 3 Results

The results achieved by the participating teams during the automated evaluation phase are shown in Table 2 and those achieved by the top three during the human evaluation phase are shown in Figure 4.

As can be observed from Table 2, the NAIS-Teacher system (Vasselli et al., 2023) attained the highest average rank on BERTScore and DialogRPT. On average, the responses were the closest to the teacher’s response, the most relevant for the

<sup>6</sup>Based on the UK government classification + Ireland.

Team	System	BERTScore			DialogRPT				Rank
		P	R	F1	U	HvR	HvM	Final	
NAIST	NAISTeacher	0.71 (9)	<b>0.71</b> (1)	<b>0.71</b> (1)	0.48 (2)	<b>0.98</b> (1)	<b>1.00</b> (1)	0.46 (2)	1.5
NBU	ADAIO	0.72 (4)	0.69 (3)	0.71 (3)	0.40 (5)	0.97 (2)	0.98 (5)	0.37 (3)	3.0
Cornell	GPT-4 <sup>(TP)</sup>	0.71 (7)	0.69 (2)	0.70 (5)	<b>0.52</b> (1)	0.86 (8)	0.98 (2)	<b>0.47</b> (1)	3.0
aaitis	S-ICL	0.72 (3)	0.69 (5)	0.70 (4)	0.40 (4)	0.92 (5)	0.98 (4)	0.36 (5)	4.5
RETUYT-InCo	OPT-2.7B	<b>0.74</b> (1)	0.68 (6)	0.71 (2)	0.38 (7)	0.90 (7)	0.96 (9)	0.35 (7)	4.5
Cornell	GPT-4	0.72 (5)	0.69 (4)	0.70 (6)	0.40 (6)	0.93 (4)	0.98 (3)	0.36 (6)	6.0
Data Science-NLP-HSG	Untrained	0.72 (6)	0.63 (8)	0.67 (8)	0.41 (3)	0.93 (3)	0.95 (10)	0.37 (4)	6.0
RETUYT-InCo	Alpaca	0.72 (2)	0.68 (7)	0.70 (7)	0.37 (8)	0.91 (6)	0.96 (7)	0.34 (8)	7.5
DT	DistilGPT2	0.67 (10)	0.62 (9)	0.64 (10)	0.36 (9)	0.75 (10)	0.96 (6)	0.29 (9)	9.5
TanTanLabs	zero-shot-with-filler	0.71 (8)	0.60 (10)	0.65 (9)	0.32 (10)	0.85 (9)	0.96 (8)	0.29 (10)	9.5
TEACHER	REFERENCE	1.00	1.00	1.00	0.37	0.86	0.99	0.32	

Table 2: Leaderboard for the evaluation phase with scores and ranks for BERTScore (P = precision, R = recall) and DialogRPT (U = updown, HvR = human vs. rand, HvM = human vs. machine)

given dialogue context, and also the most likely to be human-written. The system also achieved the second-best result on the DialogRPT updown metric, which indicated that the generated responses were likely to receive upvotes. Besides achieving the best average rank on the evaluation metrics, the system also achieved the best rank on all three criteria of pedagogical ability evaluated by human raters (see Figure 4). In particular, the responses were found to be the most helpful overall.

Table 2 further shows that the best result on the DialogRPT updown metric was achieved by the Cornell team (Hicke et al., 2023). The responses generated by GPT-4 were the most likely to receive upvotes on average (0.52) when they were submitted with a teacher prefix. However, when the team submitted the same responses *without* the prefix, they received a much lower score (0.4) and ranked 6th place on the same metric. This remarkable outcome highlighted the unanticipated sensitivity of the DialogRPT metric towards the presence or absence of a prefix.

The ADAIO system (Adigwe and Yuan, 2023)

attained the second-best average rank on both the automated evaluation phase (Table 2) and the human evaluation phase (Figure 4). The results indicated that the use of well-engineered prompts including good teaching examples (NAISTeacher, #1) and teaching approaches and goals (ADAIO, #2) resulted in a high rank on BERTScore, DialogRPT, and assessments of pedagogical ability.

It is interesting to note that the teacher’s response was ranked *lower* than the top three systems built on GPT-3 and GPT-4 (Figure 4), which contradicts the results of Tack and Piech (2022). This striking observation might be explained by some differences in the human evaluation procedure: while any native English speaker could participate in Tack and Piech (2022), only raters working in education and training could participate in the shared task. Some of these raters gave specific feedback stating that they found the non-standard language used by the teacher in the chatroom (including spelling mistakes, typos, and such) unprofessional.

For more in-depth analyses, the reader is referred to the system papers cited in this paper.

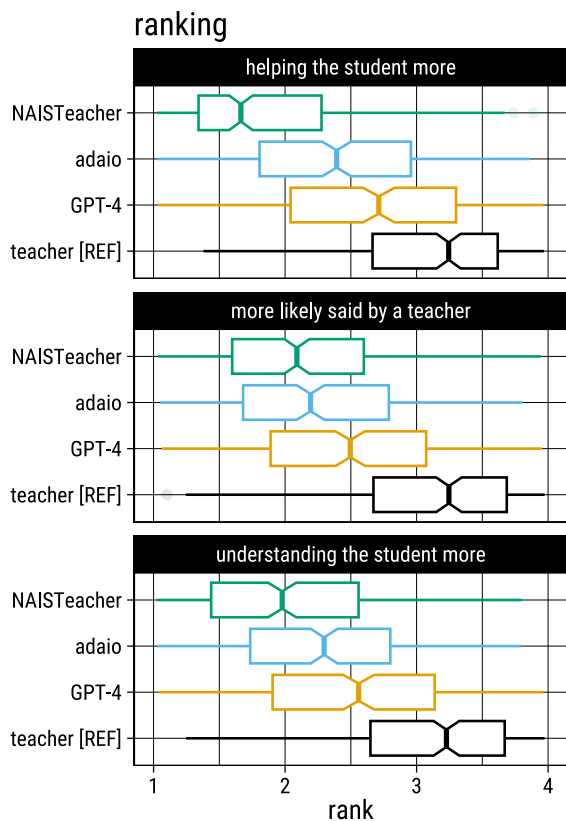


Figure 4: Ranking of the top three submissions and the teacher reference after the human evaluation phase

In these papers, the participating teams ran additional analyses and made critical observations. For example, [Baladón et al.](#) (RETUYT-InCo) observed that fine-tuned models attained better results on BERTScore, prompting attained better results on DialogRPT, and methods that combined both techniques showed competitive results across all metrics. At the same time, they found that a baseline generating “Hello” in response to every prompt achieved the best result for BERTScore precision and DialogRPT updown. [Huber et al.](#) (Data Science-NLP-HSG) found that GPT-2 – a smaller model with 124 million parameters – achieved competitive performance compared to the T5-base model. Moreover, they found that, even though they maximized BERTScore F1 as a reward function, their model scored highly in terms of the other evaluation metrics. [Vasselli et al.](#) (NAIST) noted that DialogRPT often preferred complete answers that were not very teacher-like over responses that helped the student find the answer by themselves.

## 4 Discussion

Although the inaugural shared task on generating AI teacher responses in educational dialogues can be considered a success, the results demonstrate that the evaluation of natural language generation models remains challenging. Ultimately, we would like to have at our disposal precise, valid, and – ideally – automated methods that reward machines and/or humans for their pedagogical abilities. However, we are probably still a long way from achieving this ultimate goal.

The automated metrics that currently exist are not capable of rewarding models for their ability to showcase pedagogical skills. In particular, to the best of our knowledge, there does not exist any comprehensive metric capable of evaluating whether responses are likely to be produced by a teacher, as well as whether they demonstrate understanding of what the student is saying and are helping the student. Moreover, popular automated metrics such as BERTScore and DialogRPT used in this task show a considerable sensitivity to construct-irrelevant variations, as is demonstrated by the use of a “Hello” baseline ([Baladón et al., 2023](#)) and an inclusion of the “teacher:” prefix ([Hicke et al., 2023](#)). Future editions of this task should, therefore, aim to either develop or resort to more accurate and domain-specific automated metrics as per the observations and suggestions from several competing teams ([Adigwe and Yuan, 2023](#); [Baladón et al., 2023](#); [Hicke et al., 2023](#); [Vasselli et al., 2023](#)).

Due to the lack of adequate metrics, we need to resort to manual evaluation methods in order to achieve more precise assessments. However, a typical drawback to manual evaluation is that it is very costly and time-consuming to have a sufficient number of raters evaluating *any* possible response that can be generated in the large space of possible teacher replies. Due to practical and budgetary limitations, it is challenging to organize a shared task during which any possible number of submissions can in principle be evaluated with adequately remunerated human evaluations.

What is more, data is very important in the context of real-world applications and shared tasks. Although the corpus used in this shared task is a valuable resource in our domain, some particularities of this corpus and the data sampling method also had an undeniable impact on the results. Therefore, in future editions of this shared task we should



rethink some of the current potential limitations, such as the fact that the dialogues had to be limited to 100 tokens, resulting in partial conversations; the fact that some dialogues, if extracted from the data randomly might have led to data leakage; and the fact that the dialogues did not always follow strictly role-alternating format, with some teacher turns being preceded by previous teacher utterances, rather than a student utterances.

In summary, the field of education has already been significantly changed by LLMs, whose capabilities keep improving constantly. We hope that this shared task will serve to help the scientific community better understand the current capabilities of LLMs in educational contexts. Having learned from this shared task and going forward, we hope to make its future iterations even more informative.

## 5 Conclusion

The primary goal of this shared task was to explore the potential of the current state-of-the-art NLP and AI methods in generating teacher responses in the context of real-world teacher–student interactions. A number of diverse and strong teams participated in the task and submitted outputs of their systems to the competition, and even more people expressed their interest. The teams used a variety of the state-of-the-art large language models and explored diverse prompting and fine-tuning approaches. Importantly, these results not only shed light on the current state-of-the-art on this task but also highlighted some critical limitations that should be addressed in the future.

## Acknowledgements

We thank the participants for their submissions and active involvement in this shared task. We are also grateful to them for the detailed and helpful peer reviews they provided to other shared task participants. Finally, we thank the anonymous raters on Prolific for having taken the time to provide us with additional feedback.

## References

Adaeze Adigwe and Zheng Yuan. 2023. The ADAIO System at the BEA-2023 Shared Task: Shared Task Generating AI Teacher Responses in Educational Dialogues. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications*, page to appear, Toronto, Canada. Association for Computational Linguistics.

Alexis Baladón, Ignacio Sastre, Luis Chiruzzo, and Aiala Rosá. 2023. RETUYT-InCo at BEA 2023 Shared Task: Tuning Open-Source LLMs for Generating Teacher Responses. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications*, page to appear, Toronto, Canada. Association for Computational Linguistics.

Serge Bibauw, Wim Van den Noortgate, Thomas François, and Piet Desmet. 2022. Dialogue systems for language learning: A meta-analysis. *Language Learning & Technology*, 26(1).

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Kohd, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2021. *On the Opportunities and Risks of Foundation Models*. Technical report, Stanford University, Center for Research on Foundation Models (CRFM).

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Andrew Caines, Helen Yannakoudakis, Helen Allen, Pascual Pérez-Paredes, Bill Byrne, and Paula Buttery. 2022. The Teacher-Student Chatroom Corpus version 2: More lessons, new annotation, automatic detection of sequence shifts. In *Proceedings of the 11th*

- Workshop on NLP for Computer Assisted Language Learning*, pages 23–35, Louvain-la-Neuve, Belgium. LiU Electronic Press.
- Andrew Caines, Helen Yannakoudakis, Helena Edmondson, Helen Allen, Pascual Pérez-Paredes, Bill Byrne, and Paula Buttery. 2020. The teacher-student chat-room corpus. In *Proceedings of the 9th Workshop on NLP for Computer Assisted Language Learning*, pages 10–20, Gothenburg, Sweden. LiU Electronic Press.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.
- Xiang Gao, Yizhe Zhang, Michel Galley, Chris Brockett, and Bill Dolan. 2020a. Dialogue response ranking training with large-scale human feedback data. *arXiv preprint arXiv:2009.06978*.
- Xiang Gao, Yizhe Zhang, Michel Galley, Chris Brockett, and Bill Dolan. 2020b. [Dialogue Response Ranking Training with Large-Scale Human Feedback Data](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 386–395, Online. Association for Computational Linguistics.
- Yann Hicke, Abhishek Masand, Wentao Guo, and Tushaar Gangavarapu. 2023. Assessing the efficacy of large language models in generating accurate teacher responses. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications*, page to appear, Toronto, Canada. Association for Computational Linguistics.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
- Thomas Huber, Christina Niklaus, and Siegfried Handschuh. 2023. Enhancing Educational Dialogues: A Reinforcement Learning Approach for Generating AI Teacher Responses. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications*, page to appear, Toronto, Canada. Association for Computational Linguistics.
- H. W. Kuhn. 1955. [The Hungarian method for the assignment problem](#). *Naval Research Logistics Quarterly*, 2(1-2):83–97.
- Margaret Li, Jason Weston, and Stephen Roller. 2019. [ACUTE-EVAL: Improved Dialogue Evaluation with Optimized Questions and Multi-turn Comparisons](#).
- Amin Omidvar and Aijun An. 2023. Empowering Conversational Agents using Semantic In-Context Learning. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications*, page to appear, Toronto, Canada. Association for Computational Linguistics.
- Adrien Pavao, Isabelle Guyon, Anne-Catherine Letournel, Xavier Baró, Hugo Escalante, Sergio Escalera, Tyler Thomas, and Zhen Xu. 2022. CodaLab Competitions: An open source platform to organize scientific challenges. *Technical report*.
- Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. [MAUVE: Measuring the Gap Between Neural Text and Human Text using Divergence Frontiers](#). In *Advances in Neural Information Processing Systems 34 Pre-Proceedings (NeurIPS 2021)*, pages 1–35.
- 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. *The Journal of Machine Learning Research*, 21(1):5485–5551.
- Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2022. Is Reinforcement Learning (Not) for Natural Language Processing?: Benchmarks, Baselines, and Building Blocks for Natural Language Policy Optimization. *arXiv preprint arXiv:2210.01241*.
- Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.
- Anaïs Tack and Chris Piech. 2022. [The AI Teacher Test: Measuring the Pedagogical Ability of Blender and GPT-3 in Educational Dialogues](#). In *Proceedings of the 15th International Conference on Educational Data Mining*, volume 15, pages 522–529, Durham, United Kingdom. International Educational Data Mining Society.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Stanford Alpaca: An instruction-following LLaMA model.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Justin Vasselli, Christopher Vasselli, Adam Nohejl, and Taro Watanabe. 2023. NAISTeacher: A Prompt and Rerank Approach to Generating Teacher Utterances in Educational Dialogues. In *Proceedings of the 18th*

*Workshop on Innovative Use of NLP for Building Educational Applications*, page to appear, Toronto, Canada. Association for Computational Linguistics.

Sebastian Wollny, Jan Schneider, Daniele Di Mitri, Joshua Weidlich, Marc Rittberger, and Hendrik Drachslar. 2021. [Are We There Yet? - A Systematic Literature Review on Chatbots in Education](#). *Frontiers in Artificial Intelligence*, 4:654924.

Yi-Ting Yeh, Maxine Eskenazi, and Shikib Mehri. 2021. [A Comprehensive Assessment of Dialog Evaluation Metrics](#). In *The First Workshop on Evaluations and Assessments of Neural Conversation Systems*, pages 15–33, 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*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

# The ADAIO System at the BEA-2023 Shared Task on Generating AI Teacher Responses in Educational Dialogues

**Adaeze Adigwe**

Center for Speech Technology Research  
University of Edinburgh  
United Kingdom  
A.O.I.Adigwe@sms.ed.ac.uk

**Zheng Yuan**

Istituto Italiano di Tecnologia,  
Università di Ferrara,  
Italy  
zheng.yuan@iit.it

## Abstract

This paper presents the ADAIO team’s system entry in the Building Educational Applications (BEA) 2023 Shared Task on Generating AI Teacher Responses in Educational Dialogues. The task aims to assess the performance of state-of-the-art generative models as AI teachers in producing suitable responses within a student-teacher dialogue. Our system comprises evaluating various baseline models using OpenAI GPT-3 and designing diverse prompts to prompt the OpenAI models for teacher response generation. After the challenge, our system achieved second place by employing a few-shot prompt-based approach with the OpenAI *text-davinci-003* model. The results highlight the few-shot learning capabilities of large-language models, particularly OpenAI’s GPT-3, in the role of AI teachers.

## 1 Introduction

The current success of large language models (LLMs) in generating natural language responses that are almost indistinguishable from that of a human indicates that AI systems are steps closer to passing the Turing test. Apart from being used as conversational agents, LLMs can be employed in various educational settings as described in [Kasneci et al. \(2023\)](#) including as an AI teacher to help students practice and improve. [Tack et al. \(2023\)](#) launches a shared task at the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023), called Generating AI Teacher Responses in Educational Dialogues. Inspired by [Tack and Piech \(2022\)](#), this task requires teams to develop Intelligent Tutoring Systems (ITS) that generate teacher responses in real-world teacher-student interactions. This task serves as a benchmark to gauge the capability of generative models in functioning as AI teachers.

Dialogue-based ITS face various requirements and challenges in meeting the needs of effective educational support. This entails generating factually

accurate content and ensuring educational efficacy by speaking to students in a teacher-like manner, understanding their needs, and helping them improve their understanding ([Tack and Piech, 2022](#)). However, several challenges must be addressed.

One significant challenge lies in acquiring appropriate data for training ITS, particularly real teacher-student interactions that cover various subjects. Another challenge involves developing models that can effectively capture the student’s learning style and accommodate long-range dependencies within conversational sequences. Furthermore, evaluating the quality of teacher responses is essential. The responses should not only sound natural but also demonstrate an understanding of the student’s queries and provide valuable guidance to help the student improve.

## 2 Related Work

Research on Intelligent Tutoring Systems has spanned many decades, with various proposed systems that include both text-based ([Graesser et al., 2005](#)), spoken dialogue tutoring systems ([Litman and Silliman, 2004](#)) and multi-modal systems that have been developed to improve student learning.

Earlier dialogue-based ITS were designed using rule-based cognitive modelling methods ([Aleven, 2010](#); [VanLehn et al., 2002](#)) in generating teacher responses. In recent years natural language generation (NLG) tasks generally benefited from models using sequence-to-sequence architectures ([Sutskever et al., 2014](#)). Current state-of-the-art models such as OpenAI GPT-3 ([Brown et al., 2020](#)) have shown tremendous results on a range of downstream NLG tasks such as response generation. One of the major underlying components of the language model is the transformer architecture ([Vaswani et al., 2017](#)) which increases its capacity for context awareness and long-range dependencies. Currently, the application of LLMs within the educational domain ([Bibauw et al., 2022](#); [Hendrycks](#)

et al., 2021) indicates they could improve student learning outcomes. However, their efficacy in conversational tutoring has not been fully evaluated (Tack and Piech, 2022).

On bench-marking the efficacy of LLMs in generating responses to accomplishing teaching goals, Tack and Piech (2022) investigate the suitability of these AI-teacher responses by comparing text generated by state-of-the-art models, Blender (Roller et al., 2020) and GPT-3, on real-world tutoring dialogue data. The paper comparatively analyses the responses based on a stack of evaluation methods. Furthermore, the paper suggests the following pedagogical dimensions to evaluate the AI-teacher generated responses, on its ability to *speak like a teacher, understand a student and help a student*. These dimensions form the core of the AI-teacher challenge.

### 3 Dataset

#### Teacher-Student Chatroom Corpus (TSCC)

The dataset used in this task is derived from the Teacher-Student Chatroom Corpus (TSCC) (Caines et al., 2020). The TSCC consists of 102 chatrooms where English as a second language (ESL) teachers interact with students to work on language exercises and assess students' language proficiency. From each dialogue, shorter passages limited to 100 tokens were extracted, comprising sequential turns between the teacher and student. These passages serve as data samples and end with the teacher's utterance, which acts as the reference response. The dataset follows a JSON format, including fields such as id, utterances (dialogue context), and response (teacher's ending utterance).

The dataset includes a train set of 2,747 dialogues with an average of 3.9 turns per dialogue ( $\pm 2.2$ , max=17). The dev set consists of 305 dialogues with an average of 4.0 turns ( $\pm 2.2$ , max=16), while the test set comprises 273 dialogues with an average of 2.6 turns ( $\pm 1.5$ , max=11). The response lengths in the train set range from 1 to 66 words, with an average of 9.1 words ( $\pm 8.2$ ) whereas the dev and test sets are without the response data.

## 4 System Architecture

### 4.1 Model

We conducted our experiments using OpenAI GPT-3 (Brown et al., 2020) pre-trained LLMs. Initial trials revealed that the *text-davinci-003* model produced responses that closely resembled human-like

and contextually relevant interactions, surpassing the performance of *ada*, *curie*, and *babbage*. Consequently, we predominantly employed this model for our experiments. However, considering the cost associated with utilizing the models, we opted for the *text-ada-001* model for the fine-tuning setting described below. A schematic overview of our experimental process is depicted in Figure 1.

### 4.2 Training Methods

Earlier deep-learning models would employ fine-tuning techniques to update the parameters of a pre-trained model by retraining it on new data samples from the target domain. Pre-trained LLMs such as GPT-3 and others have demonstrated the ability to utilize natural language prompts either with or without accompanying examples in performing downstream NLP tasks such as classification, summarization or generation (Brown et al., 2020; Liu et al., 2023). Within dialogue generation, fine-tuning with example data can lead to responses generated with desirable attributes or tones such as empathy, persuasion, encouragement, etc. In tutoring situations, there are attributes that make a good teacher, and we wanted to examine the ability of dialogue-based ITS to embody such characteristics. The training methods we explored include zero-shot, few-shot, and fine-tuning settings.

1. **Zero-Shot:** In this approach, we simply provided the GPT-3 Model with a modified version of **Prompt A** (see Section 4.3), without any example dialogues.
2. **Few-shot:** This approach features adding to the prompts five handpicked sample dialogues (see Table 3) from the training set. These dialogues included the *speaker-role* for each turn, i.e. *student*, and *teacher* just like in the training data. Our criteria were to choose dialogue examples with a teaching focus as defined in Caines et al. (2020). As per the teaching focus, we selected example dialogues that consisted of conversational sequences that sought to provide grammatical and lexical resources to the student while also showing aspects of discourse management and interactive communication. We replicated this approach using two language models, namely *text-ada-001* and *text-davinci-003*.
3. **Fine-tuning on the TSCC corpus:** We fine-tuned the *text-ada-001* model on the training

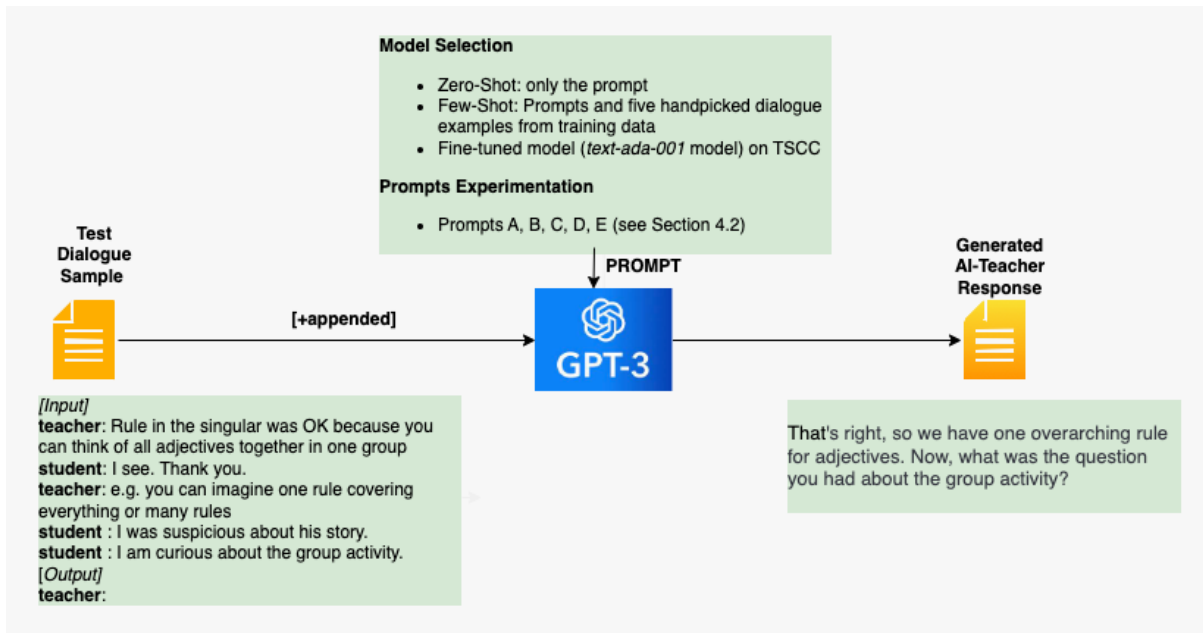


Figure 1: The framework for the proposed GPT-3 based Intelligent Tutoring System. Depending on the experimental setup, the specified prompt followed by a few handpicked dialogue examples (if applicable) is sent to the LLM (GPT-3) to generate an AI-teacher response.

data following OpenAI’s API documentation (<https://platform.openai.com/docs/guides/fine-tuning>). Our fine-tuned data consisted of approximately 95% of the training data, excluding the test data that we set aside for our internal evaluation. Afterwards, we used the fine-tuned data to prompt the model, exactly like the few-shot approach to generate the teacher responses for the test sample.

### 4.3 Prompts Engineering

In this section, we delve into the adaptability of the dialogue-based Intelligent Tutoring System (ITS) by employing prompts that experiment with various aspects, including the roles of the participants, the teaching approach adopted by the tutor, and the specific teaching goals. To achieve this, we utilized the few-shot approach, providing explicit instructions to the model regarding dialogue response generation. The prompts used, along with corresponding dialogue examples, are presented below and in Table 3.

1. **Prompt A** You will be given a dialogue chat between *a teacher and a student*, and your task is to generate a teacher response that is appropriate to the context, in which the teacher is *polite, helpful, professional, on topic, and factually correct*. The following are example dialogues with a teacher and a student.

2. **Prompt B** You will be given a dialogue chat between *a teacher and a student*, and your task is to generate a teacher response and *probe the student’s understanding in a strict manner*. The following are example dialogues with a teacher and a student.
3. **Prompt C** You will be given a dialogue chat and your task is to generate a teacher response. The following are example dialogues with a teacher and a student.
4. **Prompt D** You will be given a dialogue chat between an *English language learner and a teacher*. Your task is to generate the teacher’s response to *encourage conversational skills*. The following are example dialogues with a teacher and a student.
5. **Prompt E** You will be given a dialogue chat between *two conversational partners*. Generate the utterance that is appropriate within the dialogue context. The following are *example dialogues*.

Prompts A and B are designed to incorporate aspects of the tutor’s teaching approach, with prompt A, exhibiting more desirable attributes (adopted from Tack and Piech (2022)). In contrast, prompt B adopts a slightly different approach to probe the learner’s understanding. Prompt C takes a neutral

stance without any characteristics, putting more focus on the student-teacher roles of the dialogue participants. Prompt D attempts to generate responses that ought to focus on the learning goal - second language acquisition skills as specified in the TSCC corpus. Lastly, Prompt E removes the teacher-student roles and shifts towards dialogue participants with unspecified roles. The role tags in the few-shot examples are changed to Speaker A and Speaker B in this prompt.

#### 4.4 Implementation Details

We used the OpenAI Python library to call the GPT-3 engine to make the inferences on the test dialogues. Among the available models, we employed the top-performing *text-davinci-003* in the zero-shot and few-shot scenarios, and *text-ada-001* in the fine-tuned approach. Additionally, we compared the performance of *davinci* and *ada* in the few-shot experiments. We used the following parameters for all our experiments: *temperature=0.7*, *max tokens=100*, *top p=0.8*, *frequency penalty=0* and *presence penalty=0*. We experimented with a range of values for *max tokens* including 20, 30, 70, 100, 256. After some initial trials, we decided to go with *max-tokens=100* as it generated both a concise and relevant response most of the time. Across all our trials we kept the parameters and settings the same.

In our few-shot experimental settings, we intentionally disregarded examples samples that lacked teaching material in the reference teaching response such as turns that expressed acknowledgement, greetings or parenthetical statements, for example, conversational turns like *sure*, *okay*, *hi*, *etc*. The few-shot prompts dialogue examples were kept the same across the experiments.

## 5 Results

### 5.1 Model Selection

We randomly selected fifty samples from the training data to constitute our *internal test set* for model selection. These samples did not overlap with the few-shot dialogue examples and thus allowed us to compare the training methods listed in Section 4.2. We utilized the machine-based evaluation metric BertScore (Zhang et al., 2019) and reported the recall, precision, and F1 scores in Table 1 (all models were fed with Prompt A). The BERTScores show little variability across the models despite the apparent differences we noticed when inspect-

Models	Prec.	Rec.	F1
Zero-Shot ( <i>davinci-003</i> )	0.83	<b>0.847</b>	0.842
Few-Shot ( <i>ada-001</i> )	<b>0.848</b>	0.839	<b>0.844</b>
Few-Shot ( <i>davinci-003</i> )	0.840	0.844	0.842
Finetuned ( <i>ada-001</i> )	0.811	0.836	0.824

Table 1: BertScore evaluation of models on the internal test set

ing the generated responses. Eventually, for our final system entry, we chose the Few-Shot *davinci-003* model based on Prompt A as we believe this system generated the most meaningful responses required by the shared task. From our observation, both the few-shot and fine-tuned *ada-001 models* generated out-of-context and incoherent responses most of the time. We abstain from reporting the BertScores of models fed by Prompt B to E for the performances were consistent as shown in Table 1 and that we didn't have the resources to engage human evaluation on the quality of the generated response. Nevertheless, the generated responses piqued our interest, leading us to incorporate a few in the Appendix.

### 5.2 Shared Task Results

Table 2 presents the results of our ADAIO System (Few-Shot *davinci-003* model based on Prompt A) during the development and evaluation phases of the shared task. The numbers in parentheses represent the system's rank among the top 10 entries. The BertScore deviation observed as compared to the model selection results may be attributed to the variation in data between the reference responses in the *real test set* and the training set. Apart from BertScore, the shared task incorporates another automated dialogue evaluation metric known as DialogRPT (Gao et al., 2020). This metric assesses the generated response's performance in relation to the preceding dialogue context, considering indicators such as *updown* (the average likelihood that the response receives the most upvotes), *human vs rand* (the average likelihood that the response is contextually relevant), *human vs machine* (the average likelihood that the response is human-written rather than machine-generated), and *final* (the average/maximum) weighted ensemble score derived from all DialogRPT metrics. Our ADAIO System ranked second place after the two phases.

Phase	BERTScore			DialogRPT				
	Prec.	Rec.	F1	Updown	Human vs. Rand	Human vs. Machine	Final (avg)	Final (best)
DEV Phase	0.67(5)	0.71(1)	<b>0.69(1)</b>	0.37(5)	0.98(1)	0.99(4)	<b>0.35(2)</b>	0.71(6)
EVAL Phase	0.72(4)	0.69(3)	<b>0.71(3)</b>	0.40(5)	0.97(2)	0.98(5)	<b>0.37(3)</b>	0.65(7)

Table 2: BEA Shared Task official results of the *adaio* system

## 6 Discussion

The evaluation results from both machine and human assessments of the generated responses on the test set provide evidence of the effectiveness of LLMs, particularly GPT-3, in tutoring dialogue applications. While the dialogues in the TSCC corpus primarily concentrate on everyday speech and language usage, which proves advantageous for short conversational exchanges such as corrections, explanations, or clarifications, it is crucial to examine the GPT-3 model’s reliability in tutoring scenarios that involve longer sequences within a wider discourse context (Graesser et al., 1995). Furthermore, we perceived a limitation in relying solely on automatic evaluation metrics (as detailed in Section 5.1)

**Prompt engineering to adapt language and tone in tutoring systems** Our experiments reveal an intriguing finding where manipulating the prompt influences the tone and language of the generated response, presenting an opportunity for tutoring systems to potentially adapt to the students’ learning styles and/or teaching goals. Further research should delve into teaching instruction methods, potentially exploring the pedagogy of constructivist learning (Graesser et al., 2005) or engaging students in ill-structured exercises for productive failure (Kapur, 2008) using LLMs of this nature.

**GPT-3’s robust handling of errors and non-canonical form of language** During the data preparation phase, a manual inspection of the data revealed the presence of grammatical and spelling errors in some utterances. Additionally, since the dataset originated from chatroom text-based conversations, there were instances where mathematical symbols were used instead of natural language, such as this example utterance *Output teacher: But e.g. pleased with their visit = good idea*. It is worth noting that we did not employ any NLP processing toolkit to correct these errors or non-canonical forms in the dialogue utterances. However, despite this, the GPT-3 model could still generate appropriate responses effectively.

**LLMs’ potential in multilingual settings** In the

context of L2 acquisition, the dialogue nature in Caines et al. (2020) provides valuable opportunities for tutors to adapt to students’ native languages. Code-switching strategies as such have been found to enhance teaching, including the explanation of concepts (Köppe and Meisel, 1995), and leveraging AI tutoring systems can facilitate this process. LLMs possess multilingual capabilities that enable them to address language barriers, accommodate low-resource languages, and exhibit promising performance even on unseen languages (Yong et al., 2022). To enhance accessibility, the development and adoption of open-source multilingual models, such as BLOOM (Scao et al., 2022), should be encouraged, thereby facilitating the utilization of LLMs in educational applications across diverse linguistic contexts.

## 7 Conclusion

In this paper, we have presented our system entry to the BEA 2023 Shared Tasks on AI-teacher response generation. Our approach investigates the capability of the state-of-the-art language generative model, OpenAI GPT-3, in addressing the requirements of the AI teacher challenge outlined by Tack and Piech 2022. Through extensive experimentation utilizing zero-shot, few-shot, and fine-tuning techniques, we investigated the adaptability of the system’s responses by leveraging meticulously designed prompts and carefully selected dialogue examples that emphasize desirable teacher qualities. Our submitted system, featuring a few-shot prompt-based method, achieved 2nd place in the BEA Shared Task 2023 challenge.

## 8 Acknowledgements

The first and second authors have received funding from the European Union’s Horizon 2020 research and innovation program under the Marie Skłodowska Curie grant agreement No 859588. The authors are also indebted to Carol Figueroa for the helpful comments and feedback on the paper revision.



## References

- Vincent Aleven. 2010. Rule-based cognitive modeling for intelligent tutoring systems. *Advances in intelligent tutoring systems*, pages 33–62.
- Serge Bibauw, Wim Van den Noortgate, Thomas François, and Piet Desmet. 2022. Dialogue systems for language learning: a meta-analysis. *Language Learning & Technology*, 26(1).
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Andrew Caines, Helen Yannakoudakis, Helena Edmondson, Helen Allen, Pascual Pérez-Paredes, Bill Byrne, and Paula Buttery. 2020. The teacher-student chat-room corpus. *arXiv preprint arXiv:2011.07109*.
- Xiang Gao, Yizhe Zhang, Michel Galley, Chris Brockett, and Bill Dolan. 2020. Dialogue response ranking-training with large-scale human feedback data. In *EMNLP*.
- Arthur C Graesser, Patrick Chipman, Brian C Haynes, and Andrew Olney. 2005. Autotutor: An intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education*, 48(4):612–618.
- Arthur C Graesser, Natalie K Person, and Joseph P Magliano. 1995. Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied cognitive psychology*, 9(6):495–522.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.
- Manu Kapur. 2008. Productive failure. *Cognition and instruction*, 26(3):379–424.
- Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günemann, Eyke Hüllermeier, et al. 2023. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103:102274.
- Regina Köppe and Jürgen M Meisel. 1995. 13 code-switching in bilingual first language acquisition. *One speaker, two languages: Cross-disciplinary perspectives on code-switching*, 10:276.
- Diane Litman and Scott Silliman. 2004. Itspoke: An intelligent tutoring spoken dialogue system. In *Demonstration papers at HLT-NAACL 2004*, pages 5–8.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.
- Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M Smith, et al. 2020. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*.
- Teven Le Scao, Angela Fan, Christopher Akiki, Ellice Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.
- Anaïs Tack, Ekaterina Kochmar, Zheng Yuan, Serge Bibauw, and Chris Piech. 2023. The BEA 2023 Shared Task on Generating AI Teacher Responses in Educational Dialogues. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications*, page to appear, Toronto, Canada. Association for Computational Linguistics.
- Anaïs Tack and Chris Piech. 2022. The ai teacher test: Measuring the pedagogical ability of blender and gpt-3 in educational dialogues. *arXiv preprint arXiv:2205.07540*.
- Kurt VanLehn, Pamela W Jordan, Carolyn P Rosé, Dumisizwe Bhembe, Michael Böttner, Andy Gaydos, Maxim Makatchev, Umarani Pappuswamy, Michael Ringenberg, Antonio Roque, et al. 2002. The architecture of why2-atlas: A coach for qualitative physics essay writing. In *Intelligent Tutoring Systems: 6th International Conference, ITS 2002 Biarritz, France and San Sebastian, Spain, June 2–7, 2002 Proceedings 6*, pages 158–167. Springer.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Zheng-Xin Yong, Hailey Schoelkopf, Niklas Muenighoff, Alham Fikri Aji, David Ifeoluwa Adelani, Khalid Almubarak, M Saiful Bari, Lintang Sutawika, Jungo Kasai, Ahmed Baruwa, et al. 2022. Bloom+1: Adding language support to bloom for zero-shot prompting. *arXiv preprint arXiv:2212.09535*.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

## A Appendix

Table 3: Sample dialogues and responses in the few-shot experiment

Field	Content
Dialogue ID	0701
History	<p>student: ok,</p> <p>student: people who visited the museum after its regeneration were more pleased than the visitors in the years before</p> <p>student: I'm not sure if I can use regeneration like a synonym</p> <p>teacher: Um...good idea but not quite right here I'm afraid....maybe for e.g an area of a city or a whe industry...</p> <p>teacher: Sorry...area of a city... or whole industry</p> <p>student: yes I see</p> <p>student: because it's the same thing in Italian</p>
Ref. Responses	teacher: But e.g. pleased with their visit = good idea "
Prompt A	teacher: ' Ah yes, "regeneration" isn't quite the right word here. We want to focus on the museum specifically, so it's better to use "renovation" instead.
Prompt B	teacher: Yes, that's correct, but in this context you need to use the word 'renovation' instead of 'regeneration'. Can you explain why this is the case?
Prompt C	teacher: Yes, you can use the term "regeneration" to refer to a broader concept, such as the transformation of a whole area or industry. But in this context, it's referring specifically to the renovation of the museum.
Prompt D	teacher: Yes, that's right. You can use 'regeneration' as a synonym for 'renovation' in this context, but not for larger-scale changes like an area of a city or a whole industry.
Prompt E	SpeakerB: So, what would be a better word to use here?

Dialogue ID	1062
History	<p>student: Natural gas account for 20% of the electricity in Australia in 1980.</p> <p>teacher: yes that's great! What verb tense do you need?</p> <p>student: Natural gas made up 20% of electricity in Australia in 1980.</p> <p>teacher: Yes perfect! Good past simple too</p> <p>teacher: Account for in the past = accounted for</p> <p>student: so...Natural gas accounted for 20 % of electricity in Australia in 1980.</p> <p>student: ok..thanks</p> <p>student: I got it</p>
Ref. Responses	teacher: Yes that's great - that would look good in an answer
Prompt A	teacher: ' Great, that's exactly right! Remember, you can also use other phrases like "made up" or "constituted" to mean the same thing.
Prompt B	teacher: Great, can you explain what this sentence means in your own words?
Prompt C	teacher: Perfect! That's it.
Prompt D	teacher: Yes, that's correct! Great job!
Prompt E	SpeakerB: Great, now you try it with the other numbers in the chart.

**Continued on next page**

**Table continued from previous page**

<b>Field</b>	<b>Content</b>
Dialogue ID	0298
History	teacher: It's a bit random to be honest...if we add certain words then sometimes the singular becomes more likely e.g. the fire service is... teacher: Yes we often say the government are... teacher: a bit less formal but common yes...like we're thinking of them as a bunch of people student: ohh I didn't know that... student: But I kind of get it haha teacher: No I can see it might seem a bit strange teacher: By the way I think the plural form is less common the USA
Ref. Responses	teacher: in the US
Prompt A	teacher: That's great! Just remember that when talking about the government in the US, it's more common to use the singular form ("the government is") than the plural form ("the government are").
Prompt B	teacher: That's great to know! Can you explain to me why the plural form is less common in the US than in Britain?
Prompt C	teacher: Yes that's a good idea. It's important to be aware of the differences between different varieties of English.
Prompt D	teacher: Yes, it can be tricky! I think it's important to note that the plural form is more commonly used in Britain than in the US. But don't worry, with practice you'll get the hang of it.
Prompt E	SpeakerB: Yeah, definitely. Just remember to use "are" when talking about a group of people in the US, and "is" when talking about a single entity.

Dialogue ID	1674
History	student: plug it out teacher: nope! unplug it! teacher: A plug is also the bit at the end of a cable that you put into an electric socket teacher: Plugs can look different in different countries. For example, in Ukraine, a plug has two pins teacher: Do you know what I'm talking about? student: yes
Ref. Responses	teacher: Ok great! So you know that if you want to turn something off you unplug it, right?
Prompt A	teacher: Great! So, what do you need to do with the plug?'
Prompt B	teacher: Great! Can you explain to me what a plug is and how it works?
Prompt C	teacher: Great! So when you need to unplug something, make sure you do it carefully and don't pull on the cable itself
Prompt D	teacher: Great, can you explain what kind of plugs are used in other countries?
Prompt E	SpeakerB: So what do you need to do with it? Unplug it or plug it in?

Dialogue ID	1650
-------------	------

**Continued on next page**

**Table continued from previous page**

<b>Field</b>	<b>Content</b>
History	student: informally you hear people say e.g. So I go to the yoga class and it's so hard etc Do that people mean that they go everyday or everyweek and suffer? teacher: By the way, the structure: 'I changed my idea' is a bit weird...normal - e.g. it (the experience) changed my idea student: I changed my mind would sound better??
Ref. Responses	teacher: Ah well that would be normal in that meaning but no - i mean to describe a specific past situation
Prompt A	teacher: Yes, "I changed my mind" would be a better way of expressing it. As for the yoga class, it could mean that they attend the class on a regular basis and find it challenging.
Prompt B	teacher: Yes, "I changed my mind" would sound better. To answer your question, it depends on the context. It could mean that they go to the yoga class regularly and find it challenging, or it could mean that they went once and found it difficult.
Prompt C	teacher: Yes, "I changed my mind" would sound better. As for the yoga class, it depends on what the person is saying - it could mean they go every day, or it could mean they go occasionally and find it difficult.'
Prompt D	Yes, "I changed my mind" would sound better. It's important to use the right grammar when speaking English. But, to answer your question, it depends on the context of the conversation. People might mean that they go to the yoga class every day or every week and find it difficult, or they might just mean that it was difficult that particular time.
Prompt E	SpeakerB: Yes, 'I changed my mind' would sound better :)

# Author Index

- Adamson, David, 232  
Adigwe, Adaeze, 796  
Akinyele, Ayotomiwa, 382  
Al-Hossami, Erfan, 709  
Aleksandrova, Desislava, 518  
Alexandron, Giora, 550  
An, Aijun, 766  
Andrade, Alejandro, 137  
Ariely, Moriah, 550  
Ashok Kumar, Nischal, 247
- Baffour, Perpetual, 242  
Baghel, Shiva, 539  
Baker, Ryan, 382  
Baladn, Alexis, 756  
Ballier, Nicolas, 466  
Baraniuk, Richard, 682  
Beigman Klebanov, Beata, 550, 567  
Beinborn, Lisa, 503  
Bhargava, Muskaan, 173  
Bhattacharyya, Abhidip, 556  
Bibauw, Serge, 785  
Bitew, Semere Kiros, 598  
Bunescu, Razvan, 709  
Burstein, Jill, 394, 576
- Cai, Jon, 556  
Cai, Li, 668  
Callison-Burch, Chris, 20, 108, 382  
Cao, Jie, 556  
Chang, Kai-Wei, 668  
Chen, Benjamin, 29  
Chen, Chun-Yen, 83  
Chiruzzo, Luis, 756  
Colling, Leona, 288  
Collins, Christopher, 130  
Correnti, Richard, 275  
Crossley, Scott, 242  
Cui, Jialin, 173  
Cui, Zhiqi, 108
- Dannlls, Dana, 585  
Davidson, Sam, 83  
Deleu, Johannes, 598  
Demeester, Thomas, 598  
Demszky, Dorottya, 315, 528, 626  
Develder, Chris, 598  
Divekar, Rahul, 300
- Dmonte, Alphaeus, 404  
Dorodchi, Mohsen, 709  
Doruz, A. Seza, 598  
Doshi, Divyang, 173  
Dube, Parijat, 727  
Duenas, George, 372  
Dugan, Liam, 108  
Dwivedi, Deep, 539
- Eguchi, Masaki, 429
- Fabbri, Alex, 29  
Fang, Haishuo, 195  
Farazouli, Alexandra, 361  
Fernandez, Nigel, 247  
Fiacco, James, 232  
Fryer, Luke, 83  
Fuentes Alba, Roddy, 692  
Funakoshi, Kotaro, 184
- Gaillat, Thomas, 466  
Gangavarapu, Tushaar, 745  
Garg, Ritik, 539  
Gehring, Edward, 173  
George, Thomas, 29  
Ginsberg, Etan, 20, 108  
Gold, Christian, 352  
Gonzalez, Hannah, 108, 382  
Goodman, Noah, 315  
Guo, Wentao, 745  
Gurevych, Iryna, 195  
Gurin Schleifer, Abigail, 550
- Ha, Le An, 443  
Habermehl, Kyle, 137  
Handschuh, Siegfried, 736  
Hansen, Mark, 668  
Harik, Polina, 443  
Heck, Tanja, 44, 288  
Hellman, Scott, 137  
Hicke, Yann, 745  
Hill, Heather, 528  
Hingmire, Swapnil, 29  
Hobo, Eliza, 503  
Huber, Thomas, 736  
Huggins-Daines, David, 163
- Ide, Yusuke, 477

Jiang, Yong, 260  
 Jimenez, Sergio, 372  
 Jin, Helen, 382  
 Joanis, Eric, 163

Kaneko, Masahiro, 205  
 Katinskaia, Anisia, 488  
 Kawamura, Rina, 29  
 Knowles, Rebecca, 163  
 Kochmar, Ekaterina, 785  
 Kser, Tanja, 57  
 Kuhn, Roland, 163  
 Kulshrestha, Ritvik, 539  
 Kwako, Alexander, 668  
 Kyle, Kristopher, 429

Laarmann-Quante, Ronja, 352  
 Laflair, Geoffrey, 576  
 Lan, Andrew, 247  
 Laverghetta Jr., Antonio, 414  
 Lee, Bruce W., 1  
 Lee, Jason, 1  
 Lee, John, 448  
 Li, Bryan, 108  
 Li, Chunhui, 727  
 Li, Dawei, 260  
 Li, Irene, 29  
 Li, Jiening, 382  
 Li, Yanran, 260  
 Liang, Kai-Hui, 83  
 Liao, Tammy, 29  
 Licato, John, 414  
 Litman, Diane, 275  
 Littell, Patrick, 163  
 Liu, Chengyuan, 173  
 Liu, Fengkai, 448  
 Liu, Zhexiong, 275  
 Loem, Mengsay, 205  
 Lothian, Delaney, 163  
 Lund, Gunnar, 148

Mahajan, Khyati, 709  
 Mahamud, Mosleh, 361  
 Mallart, Cyriel, 466  
 Masand, Abhishek, 745  
 Masciolini, Arianna, 585  
 Mateus Ferro, Geral, 372  
 Matsumura, Lindsay, 275  
 Mee, Janet, 443  
 Mendez Guzman, Erick, 361

Meurers, Detmar, 44, 288  
 Miltsakaki, Eleni, 108, 382  
 Mita, Masato, 477  
 Mohania, Mukesh, 539  
 Mulcaire, Phoebe, 394

Nachman, Lama, 692  
 Nagata, Ryo, 184  
 Naismith, Ben, 394  
 Neshaei, Seyed Parsa, 57  
 Niklaus, Christina, 736  
 Nohejl, Adam, 477, 772  
 North, Kai, 404

O'reilly, Tenaha, 567  
 Oda, Mikio, 455  
 Okano, Yuki, 184  
 Okazaki, Naoaki, 205  
 Okumura, Manabu, 184  
 Okur, Eda, 692  
 Omelianchuk, Kostiantyn, 148  
 Omidvar, Amin, 766  
 Ouchi, Hiroki, 477

Panditharatne, Shehan, 83  
 Perkoff, E. Margaret, 556  
 Pham, Derek, 83  
 Piech, Chris, 785  
 Pine, Aidan, 163  
 Pouliot, Vincent, 518  
 Pouw, Charlotte, 503  
 Powell, Laurel, 709  
 Puerto, Haritz, 195  
 Purnell, Philip, 100

Quidwai, Ali, 727

Radev, Dragomir, 29  
 Ranasinghe, Tharindu, 404  
 Ren, Jiaxuan, 108, 382  
 Rietsche, Roman, 57  
 Ros, Aiala, 756  
 Ros, Carolyn, 232

Sahay, Saurav, 692  
 Samokhin, Igor, 148  
 Santandreu Calonge, David, 100  
 Sastre, Ignacio, 756  
 Saxberg, Tor, 242  
 Shang, Chenming, 260  
 Shang, Ruixuan, 173

Shardlow, Matthew, 404  
 Shea, Ryan, 83  
 Shehata, Shady, 100  
 Shi, Chufan, 260  
 Shimabukuro, Mariana, 130  
 Shin, Gyu-Ho, 72  
 Simpkin, Andrew, 466  
 Stearns, Bernardo, 466  
 Su, Xiaotian, 57  
 Suen, King Yiu, 443  
 Suhan, Michael, 567  
 Sung, Hakyung, 72  
 Suzuki, Ayaka, 119  
  
 Tack, Anas, 785  
 Takase, Sho, 205  
 Tan, Yinghua, 83  
 Teehan, Ryan, 709  
 Tessier, Marc, 163  
 Thareja, Rushil, 539  
 Thompson, Mark, 100  
 Tomikawa, Yuto, 119  
 Torkornoo, Delasie, 163  
 Tornqvist, Maximilian, 361  
  
 Ubale, Rutuja, 300  
 Upadhyay, Shriyash, 20, 108  
 Uto, Masaki, 119  
  
 Vasselli, Christopher, 772  
 Vasselli, Justin, 220, 772  
 Venant, Rmi, 466  
 Verardi, Anthony, 576  
 Volodina, Elena, 585  
 Voss, Erik, 83  
  
 Wambsganss, Thimo, 57  
  
 Wan, Yixin, 668  
 Wang, Adrian, 382  
 Wang, Elaine, 275  
 Wang, Rose, 315, 626  
 Wang, Yufang, 610  
 Wang, Zichao, 247, 682  
 Wang, Zuowei, 567  
 Watanabe, Taro, 220, 477, 772  
 Wirawarn, Pawan, 315  
  
 Xia, Lei, 610  
 Xiao, Changrong, 610  
 Xu, Dongkuan, 173  
 Xu, Sean Xin, 610  
  
 Yama, Shawn, 130  
 Yan, Vanessa, 29  
 Yancey, Kevin P., 576  
 Yaneva, Victoria, 443  
 Yangarber, Roman, 488  
 Yu Li, Jen, 466  
 Yu, Zhou, 300  
 Yuan, Xun, 83  
 Yuan, Zheng, 785, 796  
  
 Zampieri, Marcos, 404  
 Zesch, Torsten, 352  
 Zhang, Hengyuan, 260  
 Zhang, Hongyu, 382  
 Zhang, Kunpeng, 610  
 Zhang, Xuanming, 300  
 Zhao, Jieyu, 668  
 Zhou, Richard, 29  
 Zhou, Yiyun, 443  
 Zipf, Jessica, 130