BEA 2024

The 19th Workshop on Innovative Use of NLP for Building Educational Applications

Proceedings of the Workshop

June 20, 2024

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

Gold Level

BRITISH COUNCIL

CAMBRIDGE
UNIVERSITY PRESS & ASSESSMENT

CATALPA
Center of Advanced Technology for Assisted Learning and Predictive Analytics

duolingo
english test

ETS®

NBME®

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
317 Sidney Baker St. S
Suite 400 - 134
Kerrville, TX 78028
USA
Tel: +1-855-225-1962
acl@aclweb.org

ISBN 979-8-89176-100-1

Introduction

This year marks the 19th edition of the *Workshop on Innovative Use of NLP for Building Educational Applications*. As in previous years, we are happy to welcome a plethora of work on various aspects and types of educational applications – from some of the traditionally popular tasks around language learning including automated essay scoring, grammatical error detection and correction, readability assessment, and vocabulary acquisition, among others, to topics related to math and programming education, questions around empathy in teachers' responses and evaluation of teacher encouragement, fairness and explainability, bias alleviation, and ethics in AI models applied to the educational domain and many other exciting developments.

In total, we received 88 submissions, and from these, we have accepted 4 papers as talks and 34 as poster and demo presentations, for an overall acceptance rate of 43 percent. Like the rest of the NLP community, we are observing a paradigm shift, with more and more researchers applying Large Language Models (LLMs) in the context of educational applications for a variety of purposes including implementation and evaluation. A large number of papers that we have received and accepted this year investigate the topics around the integration of LLMs into educational applications. With research excellence being one of the main factors considered when making paper acceptance decisions, we hope we have also brought together a diverse program. As before, we also put a particular emphasis on multilingualism of the work included in our program, and this year BEA features work done not only on English, but also on other languages including Catalan, Danish, Dutch, Filipino, French, German, Japanese, Italian, Portuguese, Romanian, Russian, Sinhala, Spanish, and Swedish.

In addition to the diverse oral, poster, and demo presentations, this year, Alla Rozovskaya, an Assistant Professor in the Department of Computer Science at Queens College, City University of New York, will give a keynote on *Multilingual Low-Resource Natural Language Processing for Language Learning*. Furthermore, BEA 2024 has hosted two shared tasks – on *Automated Prediction of Item Difficulty and Item Response Time (APIDIRT)* and on *Multilingual Lexical Simplification Pipeline (MLSP)*. Both tasks have attracted a large number of participants, and the program includes oral presentations on the shared task descriptions from the organizers as well as extended poster sessions for shared task participants presenting their systems.

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 supporting BEA: this year, our sponsors include British Council, Cambridge University Press & Assessment, CATALPA, Duolingo English Test, Educational Testing Service, and the National Board of Medical Examiners. 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 reasonably, 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 emergency and outstanding reviewers: Michael Gringo Angelo Bayona, Jeanette Bewersdorff, Jie Cao, Scott Crossley, Sam Davidson, Kordula De Kuthy, Jasper Degraeuwe, Rujun Gao, Handoko Handoko, Michael Holcomb, Helen Jin, John Sie Yuen Lee, Hunter McNichols, Arun Balajiee Lekshmi Narayanan, Huy Viet Nguyen, Adam Nohejl, Eda Okur, Udita Patel, Martí Quixal, Manav Rathod, Alla Rozovskaya, Abhijit Suresh, Chee Wei Tan, Gladys Tyen, Justin Vasselli, Elena Volodina, ManFai Wong, Kevin Yancey, Roman Yangarber, Torsten Zesch.

Ekaterina Kochmar, MBZUAI
Marie Bexte, FernUniversität in Hagen
Jill Burstein, Duolingo
Andrea Horbach, Hildesheim University and CATALPA, FernUniversität in Hagen
Ronja Laarmann-Quante, Ruhr University Bochum
Anaïs Tack, KU Leuven, imec
Victoria Yaneva, National Board of Medical Examiners
Zheng Yuan, King’s College London

Organizing Committee

General Chair

Ekaterina Kochmar, MBZUAI

Program Chairs

Andrea Horbach, Universität Hildesheim and CATALPA, FernUniversität in Hagen
Ronja Laarmann-Quante, Ruhr University Bochum
Marie Bexte, FernUniversität in Hagen

Publication Chair

Anaïs Tack, KU Leuven, imec

Shared Tasks Chairs

Victoria Yaneva, National Board of Medical Examiners
Jill Burstein, Duolingo

Sponsorship Chair

Zheng Yuan, King's College London

Program Committee

Chairs

Ekaterina Kochmar, MBZUAI
Marie Bexte, FernUniversität in Hagen
Jill Burstein, Duolingo
Andrea Horbach, Universität Hildesheim
Ronja Laarmann-Quante, Ruhr University Bochum
Anaïs Tack, KU Leuven; imec; UCLouvain
Victoria Yaneva, National Board of Medical Examiners
Zheng Yuan, King's College London

Program Committee

Tazin Afrin, Educational Testing Service
Prabhat Agarwal, Pinterest
Erfan Al-Hossami, University of North Carolina at Charlotte
Desislava Aleksandrova, CBC/Radio-Canada
Giora Alexandron, Weizmann Institute of Science
David Alfter, UCLouvain
Fernando Alva-Manchego, Cardiff University
Jatin Ambasana, Unitedworld Institute of Technology, Karnavati University
Nico Andersen, DIPF | Leibniz Institute for Research and Information in Education
Alejandro Andrade, Pearson
Tesfa Tegegne Asfaw, Bahir Dar University
Nischal Ashok Kumar, University of Massachusetts Amherst
Berk Atil, Pennsylvania State University
Shiva Baghel, Extramarks
Rabin Banjade, University of Memphis
Stefano Banno, University of Cambridge
Michael Gringo Angelo Bayona, Trinity College Dublin
Lee Becker, Pearson
Beata Beigman Klebanov, Educational Testing Service
Lisa Beinborn, Vrije Universiteit Amsterdam
Enrico Benedetti, University of Bologna
Luca Benedetto, University of Cambridge
Jeanette Bewersdorff, FernUniversität in Hagen
Ummugul Bezirhan, Boston College, TIMSS and PIRLS International Study Center
Smita Bhattacharya, Saarland University
Abhidip Bhattacharyya, University of Massachusetts, Amherst
Serge Bibauw, UCLouvain
Robert-Mihai Botarleanu, National University of Science and Technology POLITEHNICA Bucharest
Allison Bradford, University of California, Berkeley
Ted Briscoe, MBZUAI
Jie Cao, University of Colorado
Dan Carpenter, North Carolina State University
Dumitru-Clementin Cercel, University Politehnica of Bucharest
Imran Chamieh, Hochschule Ruhr West

Jeevan Chapagain, UniversityofMemphis
Mei-Hua Chen, Department of Foreign Languages and Literature, Tunghai University
Luis Chiruzzo, Universidad de la Republica
Yan Cong, Purdue University
Mark Core, University of Southern California
Steven Coyne, Tohoku University / RIKEN
Scott Crossley, Georgia State University
Sam Davidson, University of California, Davis
Orphee De Clercq, LT3, Ghent University
Kordula De Kuthy, Universität Tübingen
Michiel De Vrindt, KU Leuven
Jasper Degraeuwe, Ghent University
Dorottya Demszky, Stanford University
Yang Deng, The Chinese University of Hong Kong
Aniket Deroy, IIT Kharagpur
Chris Develder, Ghent University
Yuning Ding, FernUniversität in Hagen
Rahul Divekar, Educational Testing Service
George Duenas, Universidad Pedagogica Nacional
Matthew Durward, University of Canterbury
Yo Ehara, Tokyo Gakugei University
Yao-Chung Fan, National Chung Hsing University
Effat Farhana, VanderbiltUniversity
Mariano Felice, University of Cambridge
Nigel Fernandez, University of Massachusetts Amherst
Michael Flor, Educational Testing Service
Jennifer-Carmen Frey, EURAC Research
Kotaro Funakoshi, Tokyo Institute of Technology
Thomas Gaillat, Rennes 2 university
Diana Galvan-Sosa, University of Cambridge
Ashwinkumar Ganesan, Amazon Alexa AI
Achyutarama Ganti, Oakland University
Rujun Gao, Texas A&M University
Ritik Garg, Extramarks Education Pvt. Ltd.
Dominik Glandorf, University of Tübingen, Yale University
Christian Gold, Fernuniversitaet Hagen
Sebastian Gombert, DIPF | Leibniz Institute for Research and Information in Education
Kiel Gonzales, University of the Philippines Diliman
Cyril Goutte, National Research Council Canada
Prasoon Goyal, The University of Texas at Austin
Pranav Gupta, Cornell University
Abigail Gurin Schleifer, Weizmann Institute of Science
Handoko Handoko, Universitas Andalas
Ching Nam Hang, Department of Computer Science, City University of Hong Kong
Jiangang Hao, Educational Testing Service
Ahatsham Hayat, University of Nebraska-Lincoln
Nicolas Hernandez, Nantes University
Nils Hjortnaes, Indiana University Bloomington
Michael Holcomb, University of Texas Southwestern Medical Center
Heiko Holz, Ludwigsburg University of Education
Sukhyun Hong, Hyperconnect, Matchgroup

Chung-Chi Huang, Frostburg State University
Chieh-Yang Huang, MetaMetrics Inc
Anna Huelsing, University of Hildesheim
Syed-Amad Hussain, Ohio State University
Catherine Ikae, Applied Machine Intelligence, Bern University of Applied Sciences, Switzerland
Joseph Marvin Imperial, University of Bath
Radu Tudor Ionescu, University of Bucharest
Suriya Prakash Jambunathan, New York University
Qinjin Jia, North Carolina State University
Helen Jin, University of Pennsylvania
Ioana Jivet, FernUniversität in Hagen
Léane Jourdan, Nantes University
Anisia Katinskaia, University of Helsinki
Elma Kerz, RWTH Aachen University
Fazel Keshtkar, St. John's University
Mamoru Komachi, Hitotsubashi University
Charles Koutcheme, Aalto University
Roland Kuhn, National Research Council of Canada
Alexander Kwako, University of California, Los Angeles
Kristopher Kyle, University of Oregon
Antonio Laverghetta Jr., Pennsylvania State University
Celine Lee, Cornell University
John Lee, City University of Hong Kong
Seolhwa Lee, Technical University of Darmstadt
Jaewook Lee, UMass Amherst
Arun Balajiee Lekshmi Narayanan, University of Pittsburgh
Yayun Li, City University of Hong Kong
Yudong Liu, Western Washington University
Zhexiong Liu, University of Pittsburgh
Naiming Liu, Rice University
Julian Lohmann, Christian Albrechts Universität Kiel
Anastassia Loukina, Grammarly Inc
Jiaying Lu, Emory University
Crisron Rudolf Lucas, UniversityCollegeDublin
Collin Lynch, NCSU
Sarah Löber, University of Tübingen
Jakub Macina, ETH Zurich
Nitin Madnani, Educational Testing Service
Jazzmin Maranan, University of the Philippines Diliman
Arianna Masciolini, University of Gothenburg
Sandeep Mathias, Presidency University
Hunter McNichols, University of Massachusetts Amherst
Jose Marie Mendoza, University of the Philippines Diliman
Amit Mishra, AmityUniversityMadhyaPradesh
Masato Mita, CyberAgent Inc.
Daniel Mora Melanchthon, Pontificia Universidad Católica de Valparaíso
Phoebe Mulcaire, Duolingo
Laura Musto, Universidad de la Republica
Ricardo Muñoz Sánchez, Språkbanken Text, Göteborgs Universitet
Farah Nadeem, LUMS
Sungjin Nam, ACT, Inc

Diane Napolitano, The Washington Post
Tanya Nazaretsky, EPFL
Kamel Nebhi, Education First
Seyed Parsa Neshaei, EPFL
Huy Nguyen, Amazon
Gebregziabihier Nigusie, Mizan-Tepi University
Christina Niklaus, University of St. Gallen
S Jaya Nirmala, National Institute of Technology Tiruchirappalli
Adam Nohejl, Nara Institute of Science and Technology
Kai North, George Mason University
Eda Okur, Intel Labs
Kostiantyn Omelianchuk, Grammarly
Amin Omidvar, PhD student at the Department of Electrical Engineering and Computer Science, York University
Benjamin Paddags, Department of Computer Science, University of Copenhagen
Ulrike Pado, HFT Stuttgart
Jeiyoon Park, Korea University
Chanjun Park, Upstage
Udita Patel, Amazon.com
Long Qin, Alibaba
Mengyang Qiu, University at Buffalo
Martí Quixal, University of Tübingen
Vatsal Raina, University of Cambridge
Manav Rathod, University of California, Berkeley
Hanumant Redkar, Goa University, Goa
Edsel Jedd Renovalles, University of the Philippines Diliman
Robert Reynolds, Brigham Young University
Saed Rezayi, National Board of Medical Examiners
Luisa Ribeiro-Flucht, University of Tuebingen
Frankie Robertson, University of Jyväskylä
Donya Rooein, Bocconi University
Aiala Rosá, Instituto de Computación, Facultad de Ingeniería, Universidad de la República
Allen Roush, University of Oregon
Alla Rozovskaya, Queens College, City University of New York
Josef Ruppenhofer, Fernuniviersität in Hagen
Horacio Saggion, Universitat Pompeu Fabra
Omer Salem, Cairo University
Nicy Scaria, Indian Institute of Science
Nils-Jonathan Schaller, Leibniz Institute for Science and Mathematics Education
Martha Shaka, University College Cork
Ashwath Shankarnarayan, New York University
Matthew Shardlow, Manchester Metropolitan University
Gyu-Ho Shin, University of Illinois Chicago
Li Siyan, Columbia University
Yixiao Song, University of Massachusetts Amherst
Mayank Soni, ADAPT Centre, Trinity College Dublin
Maja Stahl, Leibniz University Hannover
Felix Stahlberg, Google Research
Katherine Stasaski, Salesforce Research
Kevin Stowe, Educational Testing Services (ETS)
Helmer Strik, Centre for Language and Speech Technology (CLST), Centre for Language Studies

(CLS), Radboud University Nijmegen
David Strohmaier, University of Cambridge
Katsuhito Sudoh, Nara Women's University
Hakyung Sung, University of Oregon
Abhijit Suresh, Graduate Student
CheeWei Tan, NanyangTechnologicalUniversity
Zhongwei Teng, Vanderbilt University
Xiaoyi Tian, University of Florida
Gladys Tyen, University of Cambridge
Shriyash Upadhyay, University of Pennsylvania
Felipe Urrutia, Center for Advanced Research in Education
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
Elena Volodina, University of Gothenburg
Jiani Wang, East China Normal University
Taro Watanabe, Nara Institute of Science and Technology
Michael White, The Ohio State University
Alistair Willis, The Open University
Anna Winklerova, Faculty of Informatics Masaryk University
Man Fai Wong, City University of Hong Kong
Simon Woodhead, Eedi
Changrong Xiao, Tsinghua University
Kevin P. Yancey, Duolingo
Roman Yangarber, University of Helsinki
Su-Youn Yoon, EduLab
Marcos Zampieri, George Mason University
Fabian Zehner, DIPF | Leibniz Institute for Research and Information in Education
Kamyar Zeinalipour, University of Siena
Torsten Zesch, Computational Linguistics, FernUniversität in Hagen
Jing Zhang, Emory University
Yang Zhong, University of Pittsburgh
Yiyun Zhou, NBME
Jessica Zipf, University of Konstanz
Michael Zock, CNRS-LIS
Bowei Zou, Institute for Infocomm Research

Keynote Talk

Multilingual Low-Resource Natural Language Processing for Language Learning

Alla Rozovskaya
Queens College, City University of New York

Abstract: Recent studies on a wide range of NLP tasks have demonstrated the effectiveness of training paradigms that integrate large language models. However, such methods require large amounts of labeled and unlabeled data, limiting their success to a small set of well-resourced languages. This talk will discuss low-resource approaches for two language learning applications. We will begin with work on generating vocabulary exercises. We will describe an approach that does not require labeled training data and can be used to adapt the exercises to the linguistic profile of the learner. Next, we will discuss our recent work on multilingual grammatical error correction (GEC), addressing the issue of training GEC models for languages with little labeled training data, and the issue of evaluating system performance when high-quality benchmarks are lacking.

Bio: Alla Rozovskaya is an Assistant Professor in the Department of Computer Science at Queens College, City University of New York (CUNY), and a member of the Doctoral Faculty of the Computer Science and Linguistics programs at the CUNY Graduate Center. She earned her Ph.D. in Computational Linguistics at the University of Illinois at Urbana-Champaign, under the supervision of Prof. Dan Roth. Her research interests lie broadly in the area of low-resource and multilingual NLP and educational applications.

Table of Contents

How Good are Modern LLMs in Generating Relevant and High-Quality Questions at Different Bloom's Skill Levels for Indian High School Social Science Curriculum?
Nicy Scaria, Suma Chenna and Deepak Subramani 1

Synthetic Data Generation for Low-resource Grammatical Error Correction with Tagged Corruption Models
Felix Stahlberg and Shankar Kumar 11

Pillars of Grammatical Error Correction: Comprehensive Inspection Of Contemporary Approaches In The Era of Large Language Models
Kostiantyn Omelianchuk, Andrii Liubonko, Oleksandr Skurzhanskyi, Artem Chernodub, Oleksandr Korniienko and Igor Samokhin 17

Using Adaptive Empathetic Responses for Teaching English
Li Siyan, Teresa Shao, Julia Hirschberg and Zhou Yu 34

Beyond Flesch-Kincaid: Prompt-based Metrics Improve Difficulty Classification of Educational Texts
Donya Rooein, Paul Röttger, Anastassia Shaitarova and Dirk Hovy 54

Large Language Models Are State-of-the-Art Evaluator for Grammatical Error Correction
Masamune Kobayashi, Masato Mita and Mamoru Komachi 68

Can Language Models Guess Your Identity? Analyzing Demographic Biases in AI Essay Scoring
Alexander Kwako and Christopher Ormerod 78

Automated Scoring of Clinical Patient Notes: Findings From the Kaggle Competition and Their Translation into Practice
Victoria Yaneva, King Yiu Suen, Le An Ha, Janet Mee, Milton Quranda and Polina Harik 87

A World CLASSE Student Summary Corpus
Scott Crossley, Perpetual Baffour, Mihai Dascalu and Stefan Ruseti 99

Improving Socratic Question Generation using Data Augmentation and Preference Optimization
Nischal Ashok Kumar and Andrew Lan 108

Scoring with Confidence? – Exploring High-confidence Scoring for Saving Manual Grading Effort
Marie Bexte, Andrea Horbach, Lena Schützler, Oliver Christ and Torsten Zesch 119

Predicting Initial Essay Quality Scores to Increase the Efficiency of Comparative Judgment Assessmen-ts
Michiel De Vrindt, Anaïs Tack, Renske Bouwer, Wim Van Den Noortgate and Marije Lesterhuis 125

Improving Transfer Learning for Early Forecasting of Academic Performance by Contextualizing Language Models
Ahatsham Hayat, Bilal Khan and Mohammad Hasan 137

Can GPT-4 do L2 analytic assessment?
Stefano Banno, Hari Krishna Vydana, Kate Knill and Mark Gales 149

Using Program Repair as a Proxy for Language Models' Feedback Ability in Programming Education
Charles Koutcheme, Nicola Dainese and Arto Hellas 165

Automated Evaluation of Teacher Encouragement of Student-to-Student Interactions in a Simulated Classroom Discussion
Michael Ilagan, Beata Beigman Klebanov and Jamie Mikeska 182

Explainable AI in Language Learning: Linking Empirical Evidence and Theoretical Concepts in Proficiency and Readability Modeling of Portuguese
Luisa Ribeiro-Flucht, Xiaobin Chen and Detmar Meurers 199

Fairness in Automated Essay Scoring: A Comparative Analysis of Algorithms on German Learner Essays from Secondary Education
Nils-Jonathan Schaller, Yuning Ding, Andrea Horbach, Jennifer Meyer and Thorben Jansen . 210

Improving Automated Distractor Generation for Math Multiple-choice Questions with Overgenerate-and-rank
Alexander Scarlatos, Wanyong Feng, Andrew Lan, Simon Woodhead and Digory Smith 222

Identifying Fairness Issues in Automatically Generated Testing Content
Kevin Stowe 232

Towards Automated Document Revision: Grammatical Error Correction, Fluency Edits, and Beyond
Masato Mita, Keisuke Sakaguchi, Masato Hagiwara, Tomoya Mizumoto, Jun Suzuki and Kentaro Inui 251

Evaluating Vocabulary Usage in LLMs
Matthew Durward and Christopher Thomson 266

Exploring LLM Prompting Strategies for Joint Essay Scoring and Feedback Generation
Maja Stahl, Leon Biermann, Andreas Nehring and Henning Wachsmuth 283

Towards Fine-Grained Pedagogical Control over English Grammar Complexity in Educational Text Generation
Dominik Glandorf and Detmar Meurers 299

LLMs in Short Answer Scoring: Limitations and Promise of Zero-Shot and Few-Shot Approaches
Imran Chamieh, Torsten Zesch and Klaus Giebermann 309

Automated Essay Scoring Using Grammatical Variety and Errors with Multi-Task Learning and Item Response Theory
Kosuke Doi, Katsuhito Sudoh and Satoshi Nakamura 316

Error Tracing in Programming: A Path to Personalised Feedback
Martha Shaka, Diego Carraro and Kenneth Brown 330

Improving Readability Assessment with Ordinal Log-Loss
Ho Hung Lim and John Lee 343

Automated Sentence Generation for a Spaced Repetition Software
Benjamin Paddags, Daniel Hershcovich and Valkyrie Savage 351

Using Large Language Models to Assess Young Students' Writing Revisions
Tianwen Li, Zhexiong Liu, Lindsay Matsumura, Elaine Wang, Diane Litman and Richard Correnti 365

Automatic Crossword Clues Extraction for Language Learning
Santiago Berruti, Arturo Collazo, Diego Sellanes, Aiala Rosá and Luis Chiruzzo 381

Anna Karenina Strikes Again: Pre-Trained LLM Embeddings May Favor High-Performing Learners
Abigail Gurin Schleifer, Beata Beigman Klebanov, Moriah Ariely and Giora Alexandron 391

Assessing Student Explanations with Large Language Models Using Fine-Tuning and Few-Shot Learning
Dan Carpenter, Wookhee Min, Seung Lee, Gamze Ozogul, Xiaoying Zheng and James Lester 403

Harnessing GPT to Study Second Language Learner Essays: Can We Use Perplexity to Determine Linguistic Competence?
Ricardo Muñoz Sánchez, Simon Dobnik and Elena Volodina 414

BERT-IRT: Accelerating Item Piloting with BERT Embeddings and Explainable IRT Models
Kevin P. Yancey, Andrew Runge, Geoffrey LaFlair and Phoebe Mulcaire 428

Transfer Learning of Argument Mining in Student Essays
Yuning Ding, Julian Lohmann, Nils-Jonathan Schaller, Thorben Jansen and Andrea Horbach 439

Building Robust Content Scoring Models for Student Explanations of Social Justice Science Issues
Allison Bradford, Kenneth Steimel, BRIAN Riordan and Marcia Linn 450

From Miscue to Evidence of Difficulty: Analysis of Automatically Detected Miscues in Oral Reading for Feedback Potential
Beata Beigman Klebanov, Michael Suhan, Tenaha O'Reilly and Zuowei Wang 459

Findings from the First Shared Task on Automated Prediction of Difficulty and Response Time for Multiple-Choice Questions
Victoria Yaneva, Kai North, Peter Baldwin, Le An Ha, Saed Rezayi, Yiyun Zhou, Sagnik Ray Choudhury, Polina Harik and Brian Clauser 470

Predicting Item Difficulty and Item Response Time with Scalar-mixed Transformer Encoder Models and Rational Network Regression Heads
Sebastian Gombert, Lukas Menzel, Daniele Di Mitri and Hendrik Drachsler 483

UnibucLLM: Harnessing LLMs for Automated Prediction of Item Difficulty and Response Time for Multiple-Choice Questions
Ana-Cristina Rogoz and Radu Tudor Ionescu 493

The British Council submission to the BEA 2024 shared task
Mariano Felice and Zeynep Duran Karaoz 503

ITEC at BEA 2024 Shared Task: Predicting Difficulty and Response Time of Medical Exam Questions with Statistical, Machine Learning, and Language Models
Anaïs Tack, Siem Buseyne, Changsheng Chen, Robbe D'hondt, Michiel De Vrindt, Alireza Gharahighehi, Sameh Metwaly, Felipe Kenji Nakano and Ann-Sophie Noreillie 512

Item Difficulty and Response Time Prediction with Large Language Models: An Empirical Analysis of USMLE Items
Okan Bulut, Guher Gorgun and Bin Tan 522

Utilizing Machine Learning to Predict Question Difficulty and Response Time for Enhanced Test Construction
Rishikesh Fulari and Jonathan Rusert 528

Leveraging Physical and Semantic Features of text item for Difficulty and Response Time Prediction of USMLE Questions
Gummuluri Venkata Ravi Ram, Ashinee Kesanam and Anand Kumar M 534

UPN-ICC at BEA 2024 Shared Task: Leveraging LLMs for Multiple-Choice Questions Difficulty Prediction
George Duenas, Sergio Jimenez and Geral Mateus Ferro 542

Using Machine Learning to Predict Item Difficulty and Response Time in Medical Tests
Mehrdad Yousefpoori-Naeim, Shayan Zargari and Zahra Hatami 551

Large Language Model-based Pipeline for Item Difficulty and Response Time Estimation for Educational Assessments
Hariram Veeramani, Surendrabikram Thapa, Natarajan Balaji Shankar and Abeer Alwan 561

UNED team at BEA 2024 Shared Task: Testing different Input Formats for predicting Item Difficulty and Response Time in Medical Exams
Alvaro Rodrigo, Sergio Moreno-Álvarez and Anselmo Peñas 567

The BEA 2024 Shared Task on the Multilingual Lexical Simplification Pipeline
Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach and Anna Huelsing . . 571

TMU-HIT at MLSP 2024: How Well Can GPT-4 Tackle Multilingual Lexical Simplification?
Taisei Enomoto, Hwichan Kim, Tosho Hirasawa, Yoshinari Nagai, Ayako Sato, Kyotaro Nakajima and Mamoru Komachi 590

ANU at MLSP-2024: Prompt-based Lexical Simplification for English and Sinhala
Sandaru Seneviratne and Hanna Suominen 599

ISEP_Presidency_University at MLSP 2024 Shared Task: Using GPT-3.5 to Generate Substitutes for Lexical Simplification
Benjamin Dutilleul, Mathis Debaillon and Sandeep Mathias 605

Machine Translation for Lexical Complexity Prediction and Lexical Simplification
Petru Cristea and Sergiu Nisioi 610

RETUYT-INCO at MLSP 2024: Experiments on Language Simplification using Embeddings, Classifiers and Large Language Models
Ignacio Sastre, Leandro Alfonso, Facundo Fleitas, Federico Gil, Andrés Lucas, Tomás Spoturno, Santiago Góngora, Aiala Rosá and Luis Chiruzzo 618

GMU at MLSP 2024: Multilingual Lexical Simplification with Transformer Models
Dhiman Goswami, Kai North and Marcos Zampieri 627

ITEC at MLSP 2024: Transferring Predictions of Lexical Difficulty from Non-Native Readers
Anaïs Tack 635

How Good are Modern LLMs in Generating Relevant and High-Quality Questions at Different Bloom's Skill Levels for Indian High School Social Science Curriculum?

Nicy Scaria[1], Suma Dharani Chenna[1,2], Deepak Subramani[1]
[1]Computational and Data Sciences, Indian Institute of Science, India
[2]School of Computer Science and Engineering, VIT-AP University, India
{nicyscaria,deepakns}@iisc.ac.in
sumadharanichenna@gmail.com

Abstract

The creation of pedagogically effective questions is a challenge for teachers and requires significant time and meticulous planning, especially in resource-constrained economies. For example, in India, assessments for social science in high schools are characterized by rote memorization without regard to higher-order skill levels. Automated educational question generation (AEQG) using large language models (LLMs) has the potential to help teachers develop assessments at scale. However, it is important to evaluate the quality and relevance of these questions. In this study, we examine the ability of different LLMs (Falcon 40B, Llama2 70B, Palm 2, GPT 3.5, and GPT 4) to generate relevant and high-quality questions of different cognitive levels, as defined by Bloom's taxonomy. We prompt each model with the same instructions and different contexts to generate 510 questions in the social science curriculum of a state educational board in India. Two human experts used a nine-item rubric to assess linguistic correctness, pedagogical relevance and quality, and adherence to Bloom's skill levels. Our results showed that 91.56% of the LLM-generated questions were relevant and of high quality. This suggests that LLMs can generate relevant and high-quality questions at different cognitive levels, making them useful for creating assessments for scaling education in resource-constrained economies.

1 Introduction

In recent years, large language models (LLMs) have seen significant advances. They undergo training on extensive text datasets sourced from the internet and are utilized for a variety of natural language processing tasks. The introduction of OpenAI's ChatGPT and Google's Bard has made LLMs more accessible to a wider audience, enabling individuals without expertise in natural language processing (NLP) to leverage them for their everyday needs. These models are characterized by their substantial size and their ability to comprehend and produce intricate text. Through instruction finetuning, language models are calibrated to adhere to user directives (Zhang et al., 2022). In contrast to conventional language models, these LLMs possess zero-shot capabilities, allowing them to handle various tasks without specific training by simply interpreting the given instructions (Kojima et al., 2022). The educational applications of LLMs are varied and promising, covering personalized content generation, assessments, and feedback (Kasneci et al., 2023).

According to World Bank data, the teacher-pupil ratio in India's high schools is 1:29[1], compared to middle and high-income countries with an average of 1:18 and 1:13, respectively. This increases the workload on teachers and the quality of the instruction and assessment decreases. In India, subjects such as history are taught and evaluated, focusing on rote memorization (Sreekanth, 2007) with minimal emphasis on higher-order thinking skills or inquiry. Inquiry-based learning with high-quality questions fosters deep engagement and real-world connections for learners (Grant et al., 2022). Assessments aligned with Bloom's taxonomy levels (Anderson and Krathwohl, 2001), as detailed in Table 1, help educators identify learning gaps and personalize instruction, but require significant time and effort to create (Kurdi et al., 2020). Automated Educational Question Generation Systems (AEQG) have the potential to reduce this burden (Mulla and Gharpure, 2023), allowing teachers to personalize instruction and enhance student participation. This study investigates the capabilities of open source and proprietary LLMs to generate high-quality, context-aligned questions with different cognitive skills for effective assessments.

Although LLMs are capable of Natural Lan-

[1] https://data.worldbank.org

Table 1: Revised Bloom's taxonomy (Anderson and Krathwohl, 2001) in ascending order in the cognitive dimension

Bloom's level	Description
Remember	Retrieve relevant knowledge from long-term memory.
Understand	Construct meaning from instructional messages, including oral, written, and graphic communication.
Apply	Carry out or use a procedure in a given situation.
Analyze	Break material into foundational parts and determine how parts relate to one another and the overall structure or purpose.
Evaluate	Make judgments based on criteria and standards.
Create	Put elements together to form a coherent whole; reorganize into a new pattern or structure.

guage Generation (NLG) tasks, their output can have errors and inconsistencies for specific contexts. These models are also prone to hallucinations (Ji et al., 2023). These issues directly impact the quality of educational questions generated, which can vary significantly across LLMs. For this reason, evaluating the quality of these questions is important. Despite the existence of automated techniques focusing on readability and linguistic aspects, these methods do not address pedagogical aspects and question appropriateness for the given context (Amidei et al., 2018a). Therefore, expert evaluation remains essential to guarantee the quality of LLM-generated questions.

In this study, we followed a zero-shot prompting approach for question generation. We prompted LLMs to generate questions at different cognitive levels, as defined in Bloom's taxonomy, on topics covering events of the Indian independence struggle from 1857 to 1947. Using five different LLMs, we generated 510 questions in total. Two subject matter experts evaluated the generated questions based on a nine-item rubric to consider both the linguistic and pedagogical aspects of the questions (Horbach et al., 2020).

This work investigates the following research questions. *(i)* Can modern LLMs generate relevant and high-quality educational questions of different cognitive levels and follow the instructions provided in the prompt?; *(ii)* Which LLM performs the best in question generation?

Our experiments and evaluations demonstrate that the questions generated by LLMs are relevant and of good quality. These LLMs can be used for AEQG with minimal effort of the educator. Our dataset 'HistoryQ'[2] containing 510 questions evaluated by two experts and annotated with Bloom's taxonomy levels will be made available for research in the development and evaluation of AEQG systems.

2 Related Work

Traditional automated question generation (AQG) systems mainly relied on question-answering datasets before the widespread adoption of LLMs. The primary reading comprehension datasets used for question generation tasks included SQUAD (Rajpurkar et al., 2016), SQuAD 2.0 (Rajpurkar et al., 2018) and NQ (Kwiatkowski et al., 2019). One of the crowd-sourced educational datasets used for question generation tasks is SciQ (Welbl et al., 2017). LearningQ(Chen et al., 2018) and EduQG(Hadifar et al., 2023) are the other two popular datasets available for AEQG. The lack of availability of these datasets for all subjects and the human expert labor associated with creating high-quality datasets restricted the ability to develop effective AQG systems (Zhang et al., 2021). With the advent of large transformer-based pre-trained large language models, NLG tasks in recent years have improved rapidly (Zhang et al., 2022). Pre-trained and fine-tuned models such as the Text-to-Text Transfer Transformer (T5) and GPT3 were used for question generation (Nguyen et al., 2022). Leaf (Vachev et al., 2022) is a question generation developed using a pre-trained T5 model. A pre-trained T5 model (EduQG) was developed in educational text to improve the quality of the generated question (Bulathwela et al., 2023). Most AEQG systems are generic with a focus on reading comprehension or science and mathematics. AEQG research for social sciences is minimal (Bechet et al., 2022; Antoine et al., 2023). Subjects like science and mathematics tend to seek precise, quantifiable,

[2] https://github.com/nicyscaria/AEQG-SocialSciences-BloomsSkills

and objective answers. But for subjects like social sciences, the questions can be more subjective, often do not have a single correct answer, and can be interpreted differently by different people.

Many AQG systems, built by fine-tuning LLMs on specific datasets such as the ones mentioned above, often generate questions that focus on lower-order cognitive skills or simply retrieve answers directly from the context information provided (Ushio et al., 2022; Bulathwela et al., 2023). Most of the questions in EduQG(Hadifar et al., 2023) are within the first three levels of Bloom's taxonomy. These questions do not assess students' higher-order thinking abilities. Bloom's taxonomy guides educators in generating learning objectives and questions to teach and test different cognitive skills. A recent work (Sridhar et al., 2023) uses GPT4 to create course content based on Bloom's taxonomy. Although automated metrics exist to evaluate machine-generated questions, they primarily analyze linguistic aspects. In the case of educational question generation, pedagogical elements play a crucial role. Expert evaluation is necessary to understand the pedagogical aspects of machine-generated questions (Horbach et al., 2020; Steuer et al., 2021). Such evaluations are also used in student-generated questions (Moore et al., 2022).

3 Methodology

3.1 Language models and content

We chose five recent open-source and proprietary LLMs for the study. LLMs used in this study were Falcon 40B (falcon-40b-instruct), Llama 2 70B (Llama-2-7b-chat-hf), Palm 2 (chat-bison-001), GPT-3.5 (gpt-3.5-turbo-0613), and GPT-4 (gpt-4-0613). Among these, Falcon 40B is the smallest LLM with 40 billion parameters and GPT 4 is the largest (rumored, as the exact number of parameters is unknown). The questions were generated for the subject "History", covering events of the Indian independence struggle from 1857 to 1947. We used content from two chapters of the tenth grade social science textbook called *Samacheer Kalvi* (Tamil Nadu Textbook and Educational Services Corporation. State Council of Educational Research and Training, 2022) used in schools under the Indian state of Tamil Nadu's educational board. The text is in English. This content served as the context for LLMs based on the questions generated. The average length of the context was around 450 words, making it equivalent to around 600 tokens. The LLMs used had a sequence length of more than 1024 tokens to accommodate this context length and instructions. We consider 17 such contexts, so that overall nearly 500 (510, to be exact) questions are generated.

3.2 Prompt design and question generation

Each prompt had a context and instructions associated with it. The prompts were designed using techniques of pattern reframing, itemizing reframing, and assertions (instead of negations) (Mishra et al., 2022). Most Indian students, even at the tertiary level of education, are only within level B2 of the Common European Reference Framework (CEFR) for English (Council of Europe. Council for Cultural Co-operation. Education Committee. Modern Languages Division, 2001; Ravindra Babu and Shiela Mani, 2018). Therefore, additional instruction was provided in the prompt to use words within the CEFR B2 level. This approach would help students better understand the questions, thus decreasing the chances of confusion or misunderstanding arising from difficulties in comprehending the language.

We gave the same prompt to all LLMs. Each LLM had to generate six questions, one for each level in Bloom's taxonomy corresponding to the 17 contexts. Each model generated 102 questions, resulting in a total of 510 questions. The sampling temperature of an LLM typically varies between 0 and 1 in most implementations. A lower temperature results in a more deterministic output from the LLM, giving preference to the most probable predictions, while a higher temperature increases the randomness in the LLM output, resulting in less probable predictions (Hinton et al., 2015; Wang et al., 2020, 2023). A temperature value of 0.9 was used for AEQG with the LLMs to maximize the variety and diversity of the generated questions. The example of generation prompts is given in the Appendix A.1.

3.3 Human evaluation

Two experts evaluated the relevance and quality of the 510 questions based on a nine-item rubric (Table 2), a modified version of the nine-item rubric in Horbach et al.'s (2020). The two experts had subject knowledge and experience in teaching the subject social sciences and worked on question-generation tasks for multiple organizations. The experts were presented with the LLM questions in random order with only context information.

Table 2: Hierarchical nine-item rubric used to evaluate questions generated by LLMs along with the percentage agreement and Cohen's κ for each item

Rubric item	Definition
Understandable (100.00%, $\kappa = 1.00$)	Could you understand what the question is asking?
ContextRelated (100.00%, $\kappa = 1.00$)	Is the question related to the context given?
Grammatical (100.00%, $\kappa = 1.00$)	Is the question grammatically well-formed?
Clear (99.61%, $\kappa = 0.79$)	Is it clear what the question asks for?
Answerable (99.60%, $\kappa = 0.88$)	Can students answer the question?
InformationNeeded (86.80%, $\kappa = 0.73$)	What kind of information is needed to answer the question? • Information presented directly and in one place only in the text • Information presented in different parts of the text • A combination of information from the text with external knowledge • General knowledge about the topic, not from the text • The reader's feelings /judgements /... about the text • The reader's feelings/judgements/... about the text with external knowledge
Central (100.00%, $\kappa = 1.00$)	Do you think being able to answer the question is important to work on the topic covered in the context?
WouldYouUseIt (90.87%, $\kappa = 0.84$)	If you were a teacher working with that text in class, do you think you would use this question?
Bloom'sLevel (89.41%, $\kappa = 0.95$)	What is the Bloom's skill associated with the question?

They were asked to respond to each question on the rubric hierarchically from top to bottom. Seven items in the rubric were a 'yes' or 'no' response. The *InformationNeeded* item comprises six unique options that indicate what information is needed to answer the question. The questions in social sciences can be subjective and sometimes do not have a single correct answer. They can be open to interpretation. Due to this, the *InformationNeeded* contains options like 'The reader's feelings /judgements /... about the text' in addition to information derived from both the text itself and external sources. The *Bloom'sLevel* item consists of the different skills defined in Bloom's taxonomy cognitive dimension, viz., remember, understand, apply, analyze, evaluate, and create. The specifics regarding the meaning of each level of Bloom's Skill are provided in Table 1. Along with 'yes' or 'no', the option 'maybe' is also added in the *WouldYouUseIt* rubric item. In the evaluation metrics, *WouldYouUseIt* is the most subjective one.

The rubric items are structured hierarchically (Table 2), which means that if a criterion in bold font is answered with a 'no', the subsequent items in the rubric would not be considered for evaluation. For instance, if *Understandable*, *Clear*, or *Answerable* is marked 'no', the following items are not evaluated for that question and are marked as 'not applicable'. This simplifies the evaluation process.

A question is relevant and of high quality if experts say 'yes' for *Understandable*, *ContextRelated*, *Grammatical*, *Clear*, *Answerable*, and *Central* and mark 'yes' or 'maybe' for *WouldYouUseIt*. Furthermore, we utilized the *Bloom'sSkill* and *CEFRLevel* to understand whether the LLM adheres to the instructions provided in the prompt. Evaluators had to select the Bloom's level for *Bloom'sSkill* metric. We used 'Text Inspector'[3] developed by Cambridge as part of their English Profile Research (Alexopoulou, 2008) to understand the CEFR level of vocabulary used in the question. The LLM adhered to the instructions provided if the *Bloom'sSkill* label given by the evaluators matches the Bloom's

[3] https://www.englishprofile.org/wordlists/text-inspector

skill level in the prompt to the LLM and if the words are within B2 for *CEFRLevel*.

Since experts' opinions on LLM-generated questions are influenced by their writing style preferences, personal beliefs, knowledge base, and focus on detail (Amidei et al., 2018b), two inter-rater reliability measures, namely, percentage agreement and Cohen's Kappa κ (Cohen, 1960; McHugh, 2012) were used. The former is the proportion of times experts agreed on a specific rating and the latter is a robust measure that accounts for the chance agreement and provides a more accurate estimate of the true agreement between experts. Cohen's κ treats all disagreements as equal, but the disagreements cannot be considered the same for the ordinal metrics, *WillYouUseIt* and *Bloom'sLevel*. In this case, we used the quadratic weighted Cohen's κ (Cohen, 1968) instead of the simple Cohen's κ to penalize considerable disagreements more than minor disagreements.

4 Results and analysis

The percentage agreements and Cohen's κ values obtained between the two human evaluators for the nine-item rubric are given in Table 2. The percentage agreements and Cohen's κ values are calculated only for questions not labeled 'no' for the preceding rubric items in the hierarchy (marked in bold). These values indicate substantial agreement between experts on most of the metric items. Four items, *Understandable*, *ContextRelated*, *Grammatical*, and *Central* had perfect agreement.

4.1 Relevance and quality metrics

Both experts rated 100% of the generated questions as *Understandable*, *ContextRelated*, and *Grammatical*. Of these, 98.82% of the questions were rated as *Clear* and 97.84% as *Answerable*. Among the *Answerable* questions, evaluators chose one option out of the six for *InformationNeeded* item. According to the evaluators, the knowledge needed to answer 19.22% of the questions could be found in one place in the context, 18.24% from a different part of the context, and 23.33% questions needed a combination of information from the context along with external knowledge. Only 0.2% of the questions required general knowledge alone to answer, with no necessary context information. 13.73% and 10.39% of the questions required the reader's judgement about the text and the reader's judgement about the text along with external knowledge, respectively, to provide an answer. Experts rated 95.88% of the questions as *Central* to the topics covered in the respective contexts. The evaluators responded either 'yes' or 'maybe' to *WouldYouUseIt* rubric item for 91.56% of the questions. Thus, we say that the experts rated 91.56% of generated questions as relevant and high quality.

Table 3: Performance of all generated questions on different evaluation metrics

Metric	Questions (%)
Relevant & High quality	91.56%
Adherence	
• Bloom'sLevel	76.53%
• CEFRLevel	87.64%

It is observed that in the *Bloom'sLevel* metric, there is an adherence of 76.53% between the evaluators and the LLM. In the *CEFRLevel*, the adherence is 87.64% (Table 3). We are releasing our dataset, 'HistoryQ' containing 510 LLM-generated questions annotated with the nine-item metric by experts along with *CEFRLevel* for further study and analysis by the community. Examples of some relevant and high-quality questions based on Bloom's taxonomy that adhered to the instructions in the prompt are given in the Appendix A.2.

4.2 Performance of different LLMs

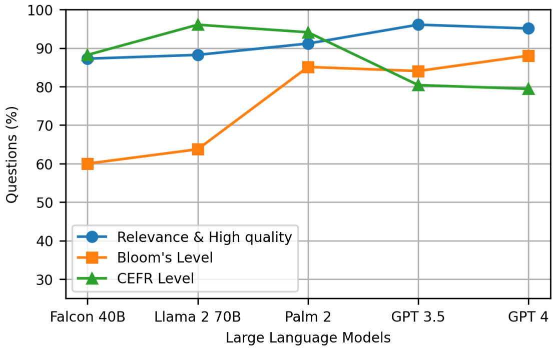


Figure 1: Performance of different LLMs on the different evaluation metrics.

The performance of the five LLMs in the AEQG task according to different evaluation criteria is summarized in Table 4. We observed that proprietary models, Palm 2, GPT 3.5, and GPT 4, which are believed to have 175 billion plus or even trillions of parameters, outperformed open-source models with 40 and 70 billion parameters in all criteria except the CEFR level adherence metric, as

Table 4: Performance of different large language models on different evaluation metrics

Metric	Falcon 40B	Llama 2 70B	Palm 2	GPT 3.5	GPT 4
Relevance & High quality	87.25%	88.24%	91.18%	96.08%	95.10%
Adherence					
• Bloom'sLevel	60.00%	63.73%	85.10%	84.04%	88.04%
• CEFRLevel	88.23%	96.07%	94.11%	80.39%	79.41%

Table 5: Precision, recall and F1 score of different large language models on Bloom's skill level compared with expert opinion

Metric	Falcon 40B	Llama 2 70B	Palm 2	GPT 3.5	GPT 4
Precision	0.60	0.65	0.85	0.84	0.87
Recall	0.60	0.66	0.86	0.86	0.88
F1 score	0.57	0.62	0.85	0.84	0.87

indicated in Figure 1.

Aligning with Bloom's taxonomy level was one of the important criteria in this study. The skill levels given by the LLM for the generated questions were compared with the ground-truth skill level labels provided by the human raters. The corresponding precision, recall, and F1 score for this task are shown in Table 5. GPT 4 outperforms other models, while Palm 2 and GPT 3.5 are in the second and third positions.

5 Conclusion

We found that 91.56% of the questions generated by different LLMs are relevant and of high quality. This indicates that LLMs can be used for AEQG with minimal effort of the educator. However, the performance varies between different LLMs. GPT 3.5 and GPT 4 generated the highest proportion of relevant and high-quality questions. In the metric of adherence to Bloom's level, GPT 4 outperformed the other models, followed by Palm 2. In contrast, the open source LLMs, Falcon 40B and Llama 2 70B, performed poorly on all metrics, except adherence to CEFR levels. This could be due to the large size of these proprietary models, which results in their ability to capture and represent complex patterns in the text data. Another interesting observation in the study was the inability of most models to generate high-quality questions at the 'Apply' and 'Create' levels of Bloom's taxonomy. GPT 3.5 and GPT 4 showed comparable performance in all criteria. Surprisingly, GPT 4 and GPT 3.5 had poor alignment with the CEFR level requested in the prompt. These models produced complex texts compared to other models.

Our research suggests that educators can leverage Palm 2, GPT 3.5, and GPT 4 to create relevant, high-quality questions of different cognitive levels defined by Bloom's taxonomy for scaling social science research in India. The LLMs must be prompted with the context in English obtained from the relevant curriculum. This approach considerably reduces the workload on teachers, especially in an under-resourced school setting where the teacher-pupil ratio is low. In addition, students can create practice tests for themselves and identify learning gaps. Expert-evaluated 'HistoryQ' could serve as a training and validation dataset for research involving the development and evaluation of AEQG models with a focus on higher-order cognitive skills.

6 Limitations

Our study required considerable time and effort from experts. Despite rigorous efforts to ensure objectivity in the evaluation through a detailed rubric and a randomized presentation of LLM-generated questions, it is important to recognize that expert evaluations can still exhibit inherent subjectivity, influenced by individual perspectives and biases. An automated system to assess the quality of machine-generated questions for their pedagogical and linguistic aspects can reduce this time and effort. This paves the way for exploring and creating high-quality automated evaluation systems. Furthermore, our study used the same prompt in different contexts for all LLMs. We did not investigate the performance of models on diverse prompts with additional information or few-shot prompting. This is another potential future direction for exploring the performance of LLMs.

References

Theodora Alexopoulou. 2008. Building new corpora for english profile. *Research Notes*, 33:15–19.

Jacopo Amidei, Paul Piwek, and Alistair Willis. 2018a. Evaluation methodologies in automatic question generation 2013-2018. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 307–317.

Jacopo Amidei, Paul Piwek, and Alistair Willis. 2018b. Rethinking the agreement in human evaluation tasks. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3318–3329.

Lorin W Anderson and David R Krathwohl. 2001. *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives: complete edition*. Addison Wesley Longman, Inc.

Elie Antoine, Hyun Jung Kang, Ismaël Rousseau, Ghislaine Azémard, Frédéric Bechet, and Géraldine Damnati. 2023. Exploring social sciences archives with explainable document linkage through question generation. In *Proceedings of the 7th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 141–151.

Frédéric Bechet, Elie Antoine, Jeremy Auguste, and Géraldine Damnati. 2022. Question generation and answering for exploring digital humanities collections. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4561–4568.

Sahan Bulathwela, Hamze Muse, and Emine Yilmaz. 2023. Scalable educational question generation with pre-trained language models. In *International Conference on Artificial Intelligence in Education*, pages 327–339. Springer.

Guanliang Chen, Jie Yang, Claudia Hauff, and Geert-Jan Houben. 2018. Learningq: a large-scale dataset for educational question generation. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 12.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Jacob Cohen. 1968. Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.

Council of Europe. Council for Cultural Co-operation. Education Committee. Modern Languages Division. 2001. *Common European framework of reference for languages: Learning, teaching, assessment*. Cambridge University Press.

SG Grant, Kathy Swan, and John Lee. 2022. *Inquiry-based practice in social studies education: Understanding the inquiry design model*. Taylor & Francis.

Amir Hadifar, Semere Kiros Bitew, Johannes Deleu, Chris Develder, and Thomas Demeester. 2023. Eduqg: A multi-format multiple-choice dataset for the educational domain. *IEEE Access*, 11:20885–20896.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Andrea Horbach, Itziar Aldabe, Marie Bexte, Oier Lopez de Lacalle, and Montse Maritxalar. 2020. Linguistic appropriateness and pedagogic usefulness of reading comprehension questions. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1753–1762.

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.

Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. 2023. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

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.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2022. Reframing Instructional Prompts to GPTk's Language. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 589–612.

Steven Moore, Huy A Nguyen, Norman Bier, Tanvi Domadia, and John Stamper. 2022. Assessing the quality of student-generated short answer questions using gpt-3. In *European conference on technology enhanced learning*, pages 243–257. Springer.

Nikahat Mulla and Prachi Gharpure. 2023. Automatic question generation: a review of methodologies, datasets, evaluation metrics, and applications. *Progress in Artificial Intelligence*, 12(1):1–32.

Huy A Nguyen, Shravya Bhat, Steven Moore, Norman Bier, and John Stamper. 2022. Towards generalized methods for automatic question generation in educational domains. In *EC-TEL*, pages 272–284. Springer.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789.

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.

KVB Ravindra Babu and K Ratna Shiela Mani. 2018. Where do we stand on cefr? an analytical study on esl learners' language proficiency. *Language in India*, 18(12).

Y Sreekanth. 2007. An analysis of question papers of different boards of examinations in social sciences. *Indian Educational Review*, 43(2):18.

Pragnya Sridhar, Aidan Doyle, Arav Agarwal, Christopher Bogart, Jaromir Savelka, and Majd Sakr. 2023. Harnessing llms in curricular design: Using gpt-4 to support authoring of learning objectives. *arXiv preprint arXiv:2306.17459*.

Tim Steuer, Leonard Bongard, Jan Uhlig, and Gianluca Zimmer. 2021. On the linguistic and pedagogical quality of automatic question generation via neural machine translation. In *EC-TEL 2021, Proceedings*, pages 289–294. Springer.

Tamil Nadu Textbook and Educational Services Corporation. State Council of Educational Research and Training. 2022. *Standard Ten, Social Science*. Directorate of School Education Tamil Nadu.

Asahi Ushio, Fernando Alva-Manchego, and Jose Camacho-Collados. 2022. Generative Language Models for Paragraph-Level Question Generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, U.A.E. Association for Computational Linguistics.

Kristiyan Vachev, Momchil Hardalov, Georgi Karadzhov, Georgi Georgiev, Ivan Koychev, and Preslav Nakov. 2022. Leaf: Multiple-choice question generation. In *European Conference on Information Retrieval*, pages 321–328. Springer.

Chi Wang, Xueqing Liu, and Ahmed Hassan Awadallah. 2023. Cost-effective hyperparameter optimization for large language model generation inference. In *International Conference on Automated Machine Learning*, pages 21–1. PMLR.

Pei-Hsin Wang, Sheng-Iou Hsieh, Shih-Chieh Chang, Yu-Ting Chen, Jia-Yu Pan, Wei Wei, and Da-Chang Juan. 2020. Contextual temperature for language modeling. *arXiv preprint arXiv:2012.13575*.

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.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2022. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Computing Surveys*.

Ruqing Zhang, Jiafeng Guo, Lu Chen, Yixing Fan, and Xueqi Cheng. 2021. A review on question generation from natural language text. *ACM TOIS*, 40(1):1–43.

A Appendix

A.1 Example prompt with a specific context

The example prompt for a specific context given to the LLMs to generate the questions is given below. All the instructions and other details remained the same for other prompts except for the context information.

Please read through the following context and instructions to create high quality questions based on the context and as per the instructions.

Context*:*

In 1857, British rule witnessed the biggest challenge to its existence. Initially, it began as a mutiny of Bengal presidency sepoys but later expanded to the other parts of India involving a large number of civilians, especially peasants. The events of 1857–58 are significant for the following reasons: 1. This was the first major revolt of armed forces accompanied by civilian rebellion. 2. The revolt witnessed unprecedented violence, perpetrated by both sides. 3. The revolt ended the role of the East India Company and the governance of the Indian subcontinent was taken over by the British Crown.

(a) Causes

1. *Annexation Policy of British India*
   In the 1840s and 1850s, more territories were annexed through two major policies: The Doctrine of Paramountcy. British claimed themselves as paramount, exercising supreme authority. New territories were annexed on the grounds that the native rulers were inept, and the Doctrine of Lapse. If a native ruler did not have male heir to the throne, the territory was to 'lapse' into British India upon the death of the ruler. Satara, Sambalpur, parts of the Punjab, Jhansi and Nagpur were annexed by the British through the Doctrine of Lapse.
2. *Insensitivity to Indian Cultural Sentiments*
   In 1806 the sepoys at Vellore mutinied against the new dress code, which prohibited Indians from wearing religious marks on their foreheads and having whiskers on their chin, while proposing to replace their turbans with a round hat. It was feared that the dress code was part of their effort to convert soldiers to Christianity. Similarly, in 1824, the sepoys at Barrackpur near Calcutta refused to go to Burma by sea, since crossing the sea meant the loss of their caste. The sepoys were also upset with discrimination in salary and promotion. Indian sepoys were paid much less than their European counterparts. They felt humiliated and racially abused by their seniors.

(b) The Revolt of 1857

The precursor to the revolt was the circulation of rumors about the cartridges of the new Enfield rifle. There was strong suspicion that the new cartridges had been greased with cow and pig fat. The cartridge had to be bitten off before loading (pork is forbidden to the Muslims and the cow is sacred to a large section of Hindus). On 29 March a sepoy named Mangal Pandey assaulted his European officer. His fellow soldiers refused to arrest him when ordered to do so. Mangal Pandey along with others were court-martialled and hanged. This only fuelled the anger and in the following days there were increasing incidents of disobedience. Burning and arson were reported from the army cantonments in Ambala, Lucknow, and Meerut.

Instructions:

1. *Create a question for each cognitive level in Bloom's taxonomy: remember, understand, apply, analyze, evaluate, and create from the context.*
2. *Ensure the questions use B2 level words or below of the Common European Framework of Reference for the English Language.*
3. *Make sure the questions relate to the students in India.*
4. *Make sure to connect events within the context while creating questions.*

A.2 Examples of LLM generated questions

Some questions generated by LLMs that are relevant, high-quality, and adhered to instructions are given along with Bloom's skill associated with the question.

- **Remember**: Name the three leaders referred to as Lal-Bal-Pal during the Swadeshi period.

- **Understand**: How did the Swadeshi movement help to promote Indian industries?
- **Apply**: How would you promote the concept of Swadeshi today, especially given the globalized world we live in?
- **Analyze**: How did the development of Swadeshi industries relate to the wider goals of the Swadeshi Movement?
- **Evaluate**: Considering the importance of self-sufficiency, do you think the boycott of foreign goods was an effective method in promoting Swadeshi industries? Provide reasons for your answer.
- **Create**: Compose a short speech or paragraph encouraging fellow students to support Swadeshi industries, drawing inspiration from the historical events mentioned.

Synthetic Data Generation for Low-resource Grammatical Error Correction with Tagged Corruption Models

Felix Stahlberg and **Shankar Kumar**
Google Research
{fstahlberg,shankarkumar}@google.com

Abstract

Tagged corruption models provide precise control over the introduction of grammatical errors into clean text. This capability has made them a powerful tool for generating pre-training data for grammatical error correction (GEC) in English. In this work, we demonstrate their application to four languages with substantially fewer GEC resources than English: German, Romanian, Russian, and Spanish. We release a new tagged-corruption dataset consisting of 2.5M examples per language that was generated by a fine-tuned PaLM 2 foundation model. Pre-training on tagged corruptions yields consistent gains across all four languages, especially for small model sizes and languages with limited human-labelled data.

1 Introduction

Grammatical error correction (GEC) is the task of correcting writing errors in text (see Bryant et al. (2023) for an overview). Neural sequence-to-sequence models, commonly used for GEC, are hard to train due to limited human-labelled data. A common strategy to mitigate data sparsity is to generate synthetic training data, but most existing methods do not generate sufficiently diverse errors. Modern GEC systems are expected to handle a broad range of errors involving grammar, spelling, word choice, punctuation and orthography. However, many existing data generation methods that employ rules or character- or word- level noising strategies, cover only a small subset of error types (Grundkiewicz et al., 2019; Grundkiewicz and Junczys-Dowmunt, 2019; Náplava and Straka, 2019; Lichtarge et al., 2019; Flachs et al., 2021). Stahlberg and Kumar (2021) improved the diversity of *model-based* data generation (Xie et al., 2018; Kiyono et al., 2019) by introducing *tagged corruption* models. Tagged corruption models are trained to generate an ungrammatical version of a clean sentence given a specific error type tag. For example, the incorrect plural "sheeps" of "sheep" (i.e. a noun inflection error – `NOUN:INFL`) would be represented in a sentence as follows (Stahlberg and Kumar, 2021):

> "`NOUN:INFL` There were a lot of sheep."
> → "There were a lot of sheeps."

In this work, we adapt the tagged corruption approach of Stahlberg and Kumar (2021) to languages with fewer GEC resources than English such as German, Spanish, Romanian, and Russian. We faced two major challenges: First, training tagged corruption models is more challenging due to training data scarcity. We mitigated this issue by leveraging the large language model PaLM 2 (Anil et al., 2023). Second, automatic error type annotation tools such as ERRANT (Felice et al., 2016; Bryant et al., 2017) for English are not available for most other languages. Therefore, we developed a multilingual annotation tool based on classification rules that apply to multiple languages and writing systems. Using our framework, we generated a new synthetic pre-training dataset with 2.5M examples per language. We demonstrate consistent gains from pre-training mT5 (Xue et al., 2021) models on our new dataset and then fine-tuning them on gold data. We achieve the largest improvements (up to 30% relative) for smaller models and languages with limited gold data. We have released the dataset and the error annotation tool to the scientific community.

2 Multilingual rule-based error type annotation

ERRANT (Felice et al., 2016; Bryant et al., 2017) is a rule-based system for English that classifies writing errors into 25 different error categories. Some ERRANT rules are specific to English and do not apply to other languages. German (Boyd, 2018)

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 11–16
June 20, 2024 ©2024 Association for Computational Linguistics

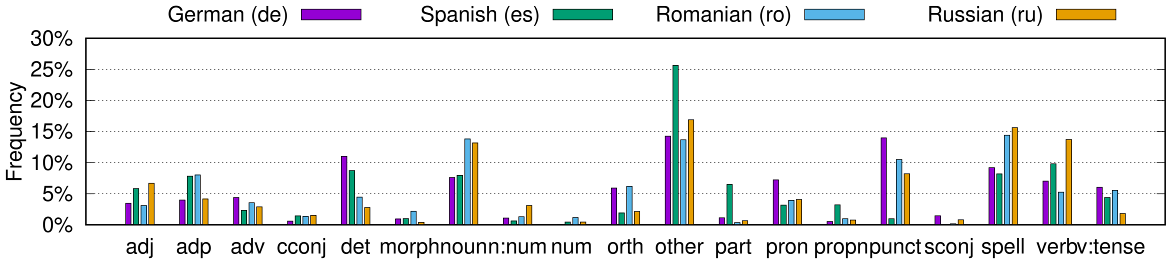


Figure 1: Development set tag distributions for German, Spanish, Romanian, and Russian.

Tag	Description
adj, adp, adv, cconj, det, part, pron, propn, sconj	Error classified by SpaCy part-of-speech (POS) tag.
morph	Morphology error.
noun	Noun or noun phrase error.
n:num	Noun number error.
num	Number error.
orth	Orthography error.
other	Unclassified error (no rule matched).
punct	Punctuation error.
spell	Spelling error according to GNU Aspell 0.60.
verb	Verb or verb phrase error.
v:tense	Verb tense error.
wo	Word order error.

Table 1: The error type tag set of our multilingual annotation tool. We use the same tag set for all languages. Rules are defined based on Aspell suggestions and SpaCy POS tags.

and Romanian (Cotet et al., 2020) versions of ERRANT have been developed, but they continue to be language-specific. Since our goal is to develop a recipe for low-resource GEC that is applicable to a large set of languages, we developed an annotation toolkit that implements a small set of general rules relying on multi-lingual NLP toolkits such as SpaCy's[1] part-of-speech (POS) tagger or GNU Aspell[2] for spelling correction. The error tag set of our tool is shown in Table 1.[3] We intentionally did not implement rules that rely on any language-specific knowledge beyond SpaCy's POS tags or Aspell suggestions. Therefore, compared to ERRANT, our tag set is more coarse-grained and less expressive. Despite the drawback, the tool's multilingual nature makes it useful for synthetic data generation across a range of languages.

3 Synthetic data generation using a tagged corruption model

Tagged corruption models are neural models that corrupt a clean sentence according to an error type tag. We adapt Stahlberg and Kumar's (2021) recipe for English data generation as follows: for each language:

1. Annotate the gold development set with error type tags using our tool from Sec. 2.
2. Compute the unigram distribution of error tags on the gold development set.
3. Sample sentences from the large clean text corpus mC4[4] (Xue et al., 2021).
4. Randomly assign an error tag to each sentence according to the tag distribution.
5. Use the tagged corruption model with temperature sampling to generate corrupted versions of the sentences. Pair them with the original sentences to build a parallel GEC dataset.
6. Filter the dataset with language identification and simple heuristics based on length offsets and edit distances.

Fig. 1 shows the tag distributions on the development set for German, Spanish, Romanian, and Russian. Our corruption model is a PaLM 2 (Anil et al., 2023) model[5] that was jointly fine-tuned on the gold training sets of all four languages. The corruption model uses the following format:

“Corrupt ⟨`lang`⟩ ⟨`tag`⟩: ⟨`clean_sentence`⟩” → “⟨`corrupted_sentence`⟩”

Fig. 2 illustrates how a training example for the corruption model is derived from the gold data. If a

[1] https://spacy.io/

[2] http://aspell.net/

[3] An open-source version of our tool is released on the dataset Github page. Please see the source code for more details about the implemented rules.

[4] https://www.tensorflow.org/datasets/catalog/c4#c4multilingual

[5] “Bison” model size available via the Google Cloud API.

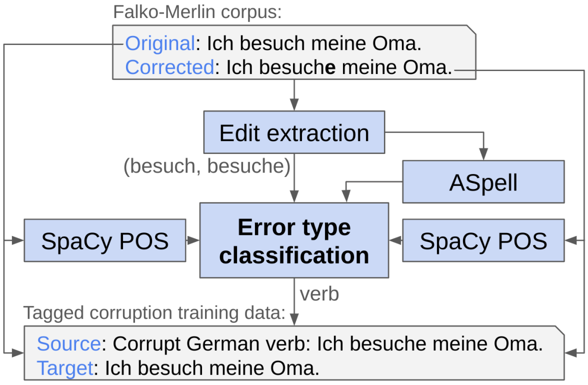

Falko-Merlin corpus:
Original: Ich besuch meine Oma.
Corrected: Ich besuche meine Oma.
Edit extraction
(besuch, besuche)
ASpell
SpaCy POS
Error type classification
SpaCy POS
verb
Tagged corruption training data:
Source: Corrupt German verb: Ich besuche meine Oma.
Target: Ich besuch meine Oma.

Figure 2: Example training instance for the tagged corruption model with a German verb error.

	de	es	ro	ru
Number of examples	2.5M			
Avg. sentence length (words)	18.9	22.0	20.8	19.1
Avg. edit distance (words)	2.8	1.9	2.3	1.5
Avg. sentence length (chars)	131.8	134.1	130.6	137.1
Avg. edit distance (chars)	5.6	5.2	3.6	4.1

Table 2: Average sentence lengths and source/target edit distances in the PRE corpus.

Language	Corpus	Train	Dev	Test
German (de)	Falko-Merlin	19.2K	2.5K	2.3K
Spanish (es)	COWS-L2H	10.1K	1.4K	1.1K
Romanian (ro)	RONACC	7.1K	1.5K	1.5K
Russian (ru)	RULEC	5.0K	2.5K	5.0K

Table 3: Number of training examples in the GOLD datasets.

sentence has multiple errors, the training example is repeated with each error tag.

Using the recipe (steps 1-6) we generated a large synthetic dataset[6] consisting of 2.5M examples per language. Table 2 lists some basic statistics of our new dataset. We will refer to this dataset as PRE.

4 Experimental setting

4.1 Gold datasets

We use the following GOLD GEC datasets for training the corruption model and for fine-tuning our GEC models: the Falko-Merlin corpus (Boyd, 2018) for German (de), the COWS-L2H corpus (Davidson et al., 2020) for Spanish (es), the RONACC corpus (Cotet et al., 2020) for Romanian (ro), and the RULEC-GEC corpus (Rozovskaya and Roth, 2019) for Russian (ru). Table 3 lists the dataset sizes.

4.2 Training setups

We train monolingual GEC models by fine-tuning the publicly available mT5 (Xue et al., 2021) checkpoints using the T5X (Roberts et al., 2023) framework on 4x4 TPUs (v3). We chose mT5 because it is available for a wide range of languages and model sizes. We use the default hyper-parameters,[7] but tune the learning rate (0.0001-0.001) and the number of training steps (1K-20K) on the respective development set. The model sizes range from *mT5-base* (580M parameters) to *mT5-xxl* (13B parameters). We compare four different training pipelines:

- GOLD: Fine-tune on the gold dataset (Sec. 4.1).
- PRE: Fine-tune on the synthetic tagged corruption dataset (Sec. 3).
- PRE→GOLD: Fine-tune first on the synthetic dataset, and then on the gold dataset.
- PRE+CLANG8→GOLD (only German and Russian): Fine-tune first on a 1:1 mix of the synthetic dataset and the CLANG8 corpus (Rothe et al., 2021), and then on the gold dataset. The CLANG8 corpus is a re-annotated version of the the language learner corpus Lang-8[8] (Mizumoto et al., 2011) available in German (114K examples) and Russian (45K examples).

5 Results

Like prior work we compute $F_{0.5}$-scores on the German, Russian, and Spanish test sets with the M2 scorer (Dahlmeier and Ng, 2012), and on the Romanian test set with Cotet et al.'s (2020) version of ERRANT.[9]

Table 4 contains the results for the three training setups for all four languages and model sizes. $F_{0.5}$-scores after training on PRE do not always surpass the GOLD baseline, which indicates that our synthetic dataset is not a replacement for human-labelled data. However, subsequent fine-tuning on GOLD after PRE consistently outperforms fine-tuning on GOLD alone, which shows the benefit of

[6] https://github.com/google-research-datasets/C4_200M-synthetic-dataset-for-grammatical-error-correction

[7] https://github.com/google-research/t5x/tree/main/t5x/examples/t5/mt5

[8] https://lang-8.com/

[9] https://github.com/teodor-cotet/errant/tree/0cb0f61af39ffb8c560ed6f92065f3b9e43e10dd

Setup	mT5-base				mT5-large				mT5-xl				mT5-xxl			
	de	es	ro	ru	de	es	ro	ru	de	es	ro	ru	de	es	ro	ru
Gold	65.6	45.9	59.4	17.3	70.6	50.5	63.2	22.8	73.5	54.8	72.4	35.0	74.9	58.1	74.4	39.5
Pre	60.8	38.6	60.7	15.9	63.9	43.6	64.0	28.4	67.3	46.6	66.1	34.7	68.4	46.4	66.6	37.8
Pre→Gold	**70.5**	**50.1**	**68.1**	**19.8**	**71.8**	**54.2**	**71.9**	**29.6**	**74.6**	**56.5**	**72.8**	**38.2**	**75.5**	**58.9**	**75.5**	**40.0**

Table 4: Test set $F_{0.5}$-scores for all four languages and model sizes. The systems highlighted in green outperform the Gold baseline.

System	German (de)	Spanish (es)	Romanian (ro)	Russian (ru)
Grundkiewicz and Junczys-Dowmunt (2019)	70.24			34.46
Náplava and Straka (2019)	73.71			50.20
Katsumata and Komachi (2020)	68.86			44.36
Cotet et al. (2020)			53.80	
Niculescu et al. (2021)			69.01	
Flachs et al. (2021)	69.24	57.32		44.72
Rothe et al. (2021)	75.96			**51.62**
Náplava et al. (2022)	73.71			50.20
Kementchedjhieva and Søgaard (2023)	73.60	55.20	68.60	49.20
This work (mT5-xxl)				
Pre→Gold	75.46	**58.89**	**75.47**	39.96
Pre+cLang8→Gold	**76.08**			44.31

Table 5: Comparison of the test set $F_{0.5}$-scores of our best systems to other results from the literature.

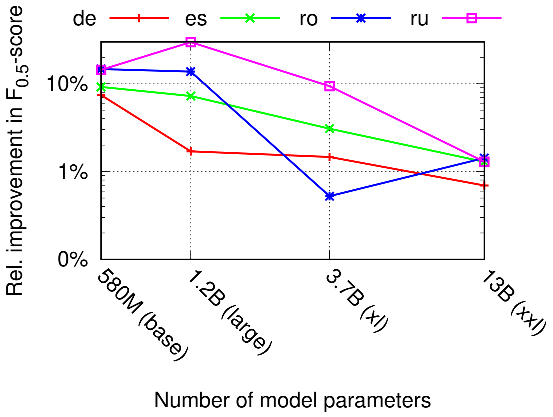


Figure 3: Relative improvements of the Pre→Gold setup over Gold-only.

adapting the model to the GEC domain before the final fine-tuning stage.

Fig. 3 shows a log-log plot of the relative improvements between the Gold baseline and the Pre→Gold setup across various model sizes. The improvements range between 0.5% and 30% depending on the language and model size. Our Pre dataset is particularly useful for small training sets (ru) and small models (left side of the plot). Grammatical error correction models deployed in practice are often small because a low latency is less disruptive for writers.

To investigate if pre-training can be further improved by adding external data, we performed experiments using the cLang8 corpus (Rothe et al., 2021). Table 6 shows that pre-training on a 1:1 mix of Pre and cLang8 outperforms pre-training on only one of them.

Setup	mT5-base		mT5-xxl	
	de	ru	de	ru
Rothe et al. (2021)	69.21	26.24	75.96	51.62
This work				
cLang8	66.39	24.58	74.83	40.37
cLang8→Gold	70.59	26.24	75.65	43.62
Pre+cLang8	69.87	25.74	74.47	**44.48**
Pre+cLang8→Gold	**72.02**	**26.39**	**76.08**	44.31

Table 6: Combining our Pre dataset with the cLang8 corpus from Rothe et al. (2021). We report $F_{0.5}$-scores on the German and Russian test sets.

Table 5 lists our best setups in relation to prior work. We advance the state-of-the-art on Spanish and Romanian and match the best published results on German despite using a relatively simple training setup (standard 2-stage fine-tuning of off-the-shelf T5 models with normal cross-entropy loss).

6 Conclusion

We have introduced a new large synthetic dataset for GEC that was generated by an LLM-based tagged corruption model in German, Spanish, Romanian, and Russian. Our dataset consists of 2.5M examples per language. Pre-training GEC models on this dataset yields consistent gains on all four languages, especially for small gold training sets and small model sizes.

7 Limitations

Even though we took into account the distribution of the error tags on the development sets for synthetic data generation, it is possible that the synthetic dataset does not capture all its error characteristics. First, our tag set is not sufficient to represent more complex inter-dependencies between error types. Second, our automated annotation tool operates on the lexical level, so clausal, sentential, or discourse level errors are not represented in the error tag set. Third, the tagged corruption model is not guaranteed to always synthesize the correct error type. Fourth, error type tags are assigned to sentences randomly, but it is sometimes not even possible to enforce an error type in a particular sentence (e.g. corrupting a sentence without a conjunction with `cconj`). Despite these limitations, we confirm Stahlberg and Kumar's (2021) findings by demonstrating the effectiveness of tagged corruption models to generate diverse synthetic training data for GEC across a range of languages.

References

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Adriane Boyd. 2018. Using Wikipedia edits in low resource grammatical error correction. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 79–84, Brussels, Belgium. 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. 2023. Grammatical Error Correction: A Survey of the State of the Art. *Computational Linguistics*, 49(3):643–701.

Teodor-Mihai Cotet, Stefan Ruseti, and Mihai Dascalu. 2020. Neural grammatical error correction for romanian. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 625–631.

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.

Sam Davidson, Aaron Yamada, Paloma Fernandez Mira, Agustina Carando, Claudia H. Sanchez Gutierrez, and Kenji Sagae. 2020. Developing NLP tools with a new corpus of learner Spanish. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 7238–7243, Marseille, France. European Language Resources Association.

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.

Simon Flachs, Felix Stahlberg, and Shankar Kumar. 2021. Data strategies for low-resource grammatical error correction. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 117–122, Online. Association for Computational Linguistics.

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.

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.

Yova Kementchedjhieva and Anders Søgaard. 2023. Grammatical error correction through round-trip machine translation. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2208–2215, Dubrovnik, Croatia. 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.

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.

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.

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.

Jakub Náplava, Milan Straka, Jana Straková, and Alexandr Rosen. 2022. Czech grammar error correction with a large and diverse corpus. *Transactions of the Association for Computational Linguistics*, 10:452–467.

Mihai Alexandru Niculescu, Stefan Ruseti, and Mihai Dascalu. 2021. Rogpt2: Romanian gpt2 for text generation. In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1154–1161.

Adam Roberts, Hyung Won Chung, Gaurav Mishra, Anselm Levskaya, James Bradbury, Daniel Andor, Sharan Narang, Brian Lester, Colin Gaffney, Afroz Mohiuddin, Curtis Hawthorne, Aitor Lewkowycz, Alex Salcianu, Marc van Zee, Jacob Austin, Sebastian Goodman, Livio Baldini Soares, Haitang Hu, Sasha Tsvyashchenko, Aakanksha Chowdhery, Jasmijn Bastings, Jannis Bulian, Xavier Garcia, Jianmo Ni, Andrew Chen, Kathleen Kenealy, Kehang Han, Michelle Casbon, Jonathan H. Clark, Stephan Lee, Dan Garrette, James Lee-Thorp, Colin Raffel, Noam Shazeer, Marvin Ritter, Maarten Bosma, Alexandre Passos, Jeremy Maitin-Shepard, Noah Fiedel, Mark Omernick, Brennan Saeta, Ryan Sepassi, Alexander Spiridonov, Joshua Newlan, and Andrea Gesmundo. 2023. Scaling up models and data with t5x and seqio. *Journal of Machine Learning Research*, 24(377):1–8.

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.

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.

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.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Pillars of Grammatical Error Correction: Comprehensive Inspection Of Contemporary Approaches In The Era of Large Language Models

Kostiantyn Omelianchuk[*]
Grammarly

Andrii Liubonko
EPAM Systems[†]

Oleksandr Skurzhanskyi
Grammarly

Artem Chernodub
Grammarly

Oleksandr Korniienko
Grammarly

Igor Samokhin
Independent Researcher[†]

Abstract

In this paper, we carry out experimental research on Grammatical Error Correction, delving into the nuances of single-model systems, comparing the efficiency of ensembling and ranking methods, and exploring the application of large language models to GEC as single-model systems, as parts of ensembles, and as ranking methods. We set new state-of-the-art performance[1] with $F_{0.5}$ scores of 72.8 on CoNLL-2014-test and 81.4 on BEA-test, respectively. To support further advancements in GEC and ensure the reproducibility of our research, we make our code, trained models, and systems' outputs publicly available.[2]

1 Introduction

Grammatical Error Correction (GEC) is the task of correcting human text for spelling and grammatical errors. There is a wide variety of GEC approaches and model architectures. In recent years, most systems have used Transformer-based architectures (Bryant et al., 2023). A current trend involves writing prompts for Large Language Models (LLMs) such as GPT-4 (OpenAI, 2023) that would generate grammatical corrections (Loem et al., 2023), (Coyne et al., 2023), (Wu et al., 2023), (Fang et al., 2023).

The varied approaches within GEC each possess unique strengths and limitations. Combining several single-model GEC systems through ensembling or ranking may smooth out their weaknesses and lead to better overall performance (Susanto et al., 2014). Even quite simple ensembling methods, such as majority voting (Tarnavskyi et al., 2022) or logistic regression (Qorib et al., 2022), may work surprisingly well. Combining single-model systems is also often straightforward from an implementation perspective. Because only the outputs of the models are required for many ensembling algorithms, there is no need to retrain models or perform inference passes iteratively. A further review of related work is presented in the end and near the descriptions of considered methods.

Our contributions are the following:

1. Comprehensive comparison of GEC methods. We reproduce, evaluate, and compare the most promising existing methods in GEC, both single-model systems and ensembles. We show that usage of ensembling methods is crucial to obtain state-of-the-art performance in GEC.

2. Establishing new state-of-the-art baselines. We show that simple ensembling by majority vote outperforms more complex approaches and significantly boosts performance. We push the boundaries of GEC quality and achieve new state-of-the-art results on the two most common GEC evaluation datasets: $F_{0.5} = 72.8$ on CoNLL-2014-test and $F_{0.5} = 81.4$ on BEA-test.

3. Exploring the application of LLMs for GEC. We thoroughly investigate different scenarios for leveraging large language models (LLMs) for GEC: 1) as single-model systems in a zero-shot setting, 2) as fine-tuned single-model systems, 3) as single-model systems within ensembles, and 4) as a combining algorithm for ensembles. To the best of our knowledge, we are the first to explore using GPT-4 to rank GEC edits, which contributes to a notable improvement in the Recall of ensemble systems.

4. Commitment to open science. In a move toward fostering transparency and encouraging further research, we open-source all our models, their outputs on evaluation datasets, and the accompanying code.[2] This ensures the reproducibility of our work and provides a foundation for future advancements in the field.

[*] Corresponding author: kostiantyn.omelianchuk@grammarly.com.

[†] The work was carried out while working at Grammarly.

[1] https://nlpprogress.com/english/grammatical_error_correction.html (Accessed 10 March 2024).

[2] https://github.com/grammarly/pillars-of-gec

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 17–33
June 20, 2024 ©2024 Association for Computational Linguistics

2 Data for Training and Evaluation

We use the following GEC datasets for training models (Table 1):

1. **Lang-8**, an annotated dataset from the Lang-8 Corpus of Learner English (Tajiri et al., 2012);

2. **NUCLE**, the National University of Singapore Corpus of Learner English (Dahlmeier et al., 2013);

3. **FCE**, the First Certificate in English dataset (Yannakoudakis et al., 2011);

4. **W&I**, the Write & Improve Corpus (Bryant et al., 2019) (also known as BEA-Train). We also use a larger synthetic version of Lang-8 with target sentences produced by the T5 model (Raffel et al., 2020);

5. **cLang-8** (Rothe et al., 2021), and synthetic data based on two monolingual datasets;

6. **Troy-1BW** (Tarnavskyi et al., 2022), produced from the One Billion Word Benchmark (Chelba et al., 2014);

7. **Troy-Blogs** (Tarnavskyi et al., 2022), produced from the Blog Authorship Corpus (Schler et al., 2006).

#	Dataset	Part	# sent.	# tokens	% edits
1	Lang-8	Train	1.04M	11.86M	42
2	NUCLE	Train	57.0k	1.16M	62
		Test	1.3k	30k	90
3	FCE	Train	28.0k	455k	62
4	W&I + LOCNESS	Train	34.3k	628.7k	67
		Dev	4.4k	85k	64
		Test	4.5k	62.5k	N/A
5	cLang-8	Train	2.37M	28.0M	58
6	Troy-1BW	Train	1.2M	30.88M	100
7	Troy-Blogs	Train	1.2M	21.49M	100

Table 1: Statistics of GEC datasets used in this work for training and evaluation.

For evaluation, we use current standard evaluation sets for the GEC domain: the test set from the CoNLL-2014 GEC Shared Task (Ng et al., 2014), and the dev and test components of the W&I + LOCNESS Corpus from the BEA-2019 GEC Shared Task (BEA-dev and BEA-test) (Bryant et al., 2019). For BEA-test, submissions were made through the current competition website.[3] For each dataset, we report Precision, Recall, and $F_{0.5}$ scores. To ensure an apples-to-apples comparison with previously reported GEC results, we evaluate CONLL-2014-test with M2scorer (Dahlmeier and Ng, 2012), and BEA-dev with ERRANT (Bryant et al., 2017).

3 Single-Model Systems

3.1 Large Language Models

We investigate the performance of open-source models from the LLaMa-2 family (Touvron et al., 2023), as well as two proprietary models: GPT-3.5 (Chat-GPT) and GPT-4 (OpenAI, 2023). For LLaMa, we work with four models: LLaMa-2-7B, LLaMa-2-13B, Chat-LLaMa-2-7B, and Chat-LLaMa-2-7B. We use two LLaMa-2 model sizes: 7B and 13B. If the model is pre-trained for instruction following (Ouyang et al., 2022), it is denoted as "Chat-" in the model's name.

Chat-GPT and GPT-4 are accessed through the Microsoft Azure API. We use versions *gpt-3.5-turbo-0613* and *gpt-4-0613*, respectively.

We explore two scenarios for performing GEC using LLMs: zero-shot prompting (denoted as "ZS") and fine-tuning (denoted as "FT").

3.1.1 Zero-Shot Prompting

In recent studies dedicated to prompting LLMs for GEC, it was shown that LLM models tend to produce more fluent rewrites (Coyne et al., 2023). At the same time, performance measured by automated metrics such as MaxMatch (Dahlmeier and Ng, 2012) or ERRANT has been identified as inferior. We frequently observed that these automated metrics do not always correlate well with human scores. This makes LLMs used in zero-shot prompting mode potentially attractive, especially in conjunction with other systems in an ensemble.

For the Chat-LLaMa-2 models, we use a two-tiered prompting approach that involves setting the system prompt *"You are a writing assistant. Please ensure that your responses consist only of corrected texts."* to provide the context to direct the model focus toward GEC task. Then, we push the following instruction prompt to direct the model's focus toward the GEC task:

```
Fix grammatical errors for the following text.
```

Temperature is set to 1. For Chat-GPT and GPT-4 models, we employ a function-calling API with the "required" parameter. This guides the LLM to more accurately identify and correct any linguistic errors within the text or replicate the input text if it was already error-free, thus ensuring consistency in the models' responses. The instruction prompt for GPT models is:

[3] https://codalab.lisn.upsaclay.fr/competitions/4057

#	System	CoNLL-2014-test			BEA-dev			BEA-test		
		Precision	Recall	$F_{0.5}$	Precision	Recall	$F_{0.5}$	Precision	Recall	$F_{0.5}$
1	Chat-LLaMa-2-7B-ZS	42.9	47.3	43.7	19.1	34.1	21.0	-	-	-
2	Chat-LLaMa-2-13B-ZS	49.1	56.1	50.4	30.6	45.0	32.7	-	-	-
3	GPT-3.5-ZS	56.2	57.7	56.5	37.4	50.6	39.4	-	-	-
4	GPT-3.5-CoT-ZS	56.0	**58.7**	56.5	36.4	**50.8**	38.5	-	-	-
5	GPT-4-ZS	**59.0**	55.4	**58.2**	**42.5**	45.0	**43.0**	-	-	-
6	Chat-LLaMa-2-7B-FT	75.5	46.8	67.2	58.3	46.0	55.3	72.3	67.4	71.2
7	Chat-LLaMa-2-13B-FT	77.3	45.6	**67.9**	59.8	46.1	56.4	74.6	67.8	73.1
8	T5-11B	70.9	**56.5**	67.5	60.9	**51.1**	**58.6**	73.2	**71.2**	72.8
9	UL2-20B	73.8	50.4	67.5	60.5	48.6	57.7	75.2	70.0	74.1
10	GECToR-2024	75.0	44.7	66.0	64.6	37.2	56.3	77.7	59.0	73.1
11	CTC-Copy	72.6	47.0	65.5	58.3	38.0	52.7	71.7	59.9	69.0
12	EditScorer	**78.5**	39.4	65.5	**67.3**	36.1	57.4	**81.0**	56.1	**74.4**

Table 2: All single-model systems evaluated on CoNLL-2014-test, BEA-dev, and BEA-test datasets.

```
Fix all mistakes in the text (spelling, punctuation,
grammar, etc). If there are no errors, respond with
the original text.
```

Additionally, we employ a form of the chain-of-thought (CoT) prompting (Wei et al., 2022), which involves requesting reasoning from the model before it makes corrections by means of function calling.

3.1.2 Fine-tuning the Large Language Models

Fine-tuning is a mainstream method for knowledge transfer. Since we have several available annotated GEC datasets, they may be used to fine-tune LLMs (Zhang et al., 2023b; Kaneko and Okazaki, 2023).

We use three datasets for fine-tuning — NUCLE, W&I, and cLang-8 (Table 1) — as they are commonly used in recent GEC research (Zhang et al., 2023b; Kaneko and Okazaki, 2023; Loem et al., 2023). We varied the datasets and their shares to find the best combination.

We use the Transformers library[4] to conduct 1000–1200 updates with 250 warm-up steps, a batch size of 8, and a learning rate of $1e-5$. We fine-tune only LLaMA-2 models on next token prediction task, both autocomplete and instruction-following pre-trained versions (denoted as "Chat-"). For the Chat-LLaMA-2 models, we use the following prompt:

```
Rewrite the following text to make it grammatically
correct.
[Input text]
Result:
[Output text]
```

Additionally, we perform an ablation study on the models' size and the usefulness of the instructions (Appendix D, Table 11). Not surprisingly, our results indicate that instructions work better for "Chat" versions of models.

3.2 Sequence-to-Sequence models

In a sequence-to-sequence approach, GEC is considered a machine translation task, where errorful sentences correspond to the source language, and error-free sentences correspond to the target language (Grundkiewicz et al., 2019; Kiyono et al., 2019). In this work, we investigate two powerful Transformer-based Seq2Seq models: the open-sourced "T5-11B" (Rothe et al., 2021), and "UL2-20B", the instruction-tuned version of FLAN (Tay et al., 2022).

T5-11B is fine-tuned on W&I + LOCNESS train data for 500 updates with batch size 256 and a learning rate of $1e-4$. UL2-20B is fine-tuned on W&I + LOCNESS train data for 300 updates with batch size 16 and a learning rate of $5e-5$.

3.3 Edit-based Systems

Edit-based GEC systems produce explicit text changes, restoring error-free language from the errorful source text. Usually, such systems are based on encoder-only architectures and are non-autoregressive; therefore, they are less resource-consuming and more attractive for productization. In this work, we consider three publicly available open-source edit-based systems for GEC: GECToR, CTC-Copy, and EditScorer.

GECToR[5] (Omelianchuk et al., 2020), (Tarnavskyi et al., 2022) is a family of non-autoregressive sequence tagging GEC systems. The concept revolves around training Transformer-based, encoder-only models to generate corrective edits.

CTC-Copy[6] (Zhang et al., 2023a) is another non-autoregressive text editing approach. It uses Con-

[4] https://github.com/huggingface/transformers

[5] https://github.com/MaksTarnavskyi/gector-large

[6] https://github.com/yzhangcs/ctc-copy

System name	CoNLL-2014-test Precision	 Recall	 $\mathbf{F_{0.5}}$	BEA-test Precision	 Recall	 $\mathbf{F_{0.5}}$
GECToR-RoBERTa$^{(L)}$ (Tarnavskyi et al., 2022)	70.1	42.7	62.2	80.6	52.3	72.7
GECToR-FT-Stage-I	**75.2**	44.1	65.9	**78.1**	57.7	72.9
GECToR-FT-Stage-II (GECToR-2024)	75.0	**44.7**	**66.0**	77.7	**59.0**	**73.1**

Table 3: GECToR fine-tuning experiments. We compare the performance of our fine-tuned model after stage I and stage II to the initial off-the-shelf model as a baseline.

nectionist Temporal Classification (CTC) (Graves et al., 2006) initially developed for automatic speech recognition and introduces a novel text editing method by modeling the editing process with latent CTC alignments. This allows more flexible editing operations to be generated.

EditScorer[7] (Sorokin, 2022) splits GEC into two steps: generating and scoring edits. We consider it a single-model system approach because all edits are generated by a single-model system.

We also attempt to reproduce the Seq2Edit approach (Stahlberg and Kumar, 2020), (Kaneko and Okazaki, 2023), but fail to achieve meaningful results. Please find more details in Appendix B.

For GECToR, we use the top-performing model, GECToR-RoBERTa$^{(L)}$ (Tarnavskyi et al., 2022). Since this model was not trained on cLang-8 data, we additionally fine-tune it on a mix of cLang-8, BEA, Troy-1BW, and Troy-Blogs data. We leverage a multi-stage fine-tuning approach from (Omelianchuk et al., 2020). In stage I, a mix of cLang-8, W&I + LOCNESS train (BEA-train), Troy-1BW, and Troy-Blogs datasets is used for fine-tuning; in stage II, the high-quality W&I + LOCNESS train dataset is used to finish the training. During stage I, we fine-tune the model for 5 epochs, early-stopping after 3 epochs, with each epoch equal to 10000 updates and a batch size of 256. During stage II, we further fine-tune the model for 4 epochs, with each epoch equal to 130 updates. The full list of hyperparameters for fine-tuning can be found in Appendix D, Table 7. We refer to this new, improved GECToR model as GECToR-2024.

For CTC-Copy, we use the official code[6] with the RoBERTa encoder to train the English GEC model.

For EditScorer, we use the open-sourced code[7] for GECToR-XLNet$^{(L)}$ option from (Tarnavskyi et al., 2022) to sample possible edits and stagewise decoding with the RoBERTa-Large encoder to rescore them.

3.4 Single-Model Systems Results

The performance of single-model GEC systems is presented in Table 2.

We see that all zero-shot approaches considered have $F_{0.5}$ scores lower than 60 on the CoNLL-2014-test dataset, which we assume to be a lower bound on satisfactory GEC quality. They all suffer from an overcorrecting issue (Fang et al., 2023), (Wu et al., 2023) that leads to poor Precision and inferior $F_{0.5}$ scores. Notably, GPT models show consistently better results compared to LLaMa. Implementing the chain-of-thought approach doesn't improve the quality.

Among the remaining approaches — LLMs with fine-tuning, sequence-to-sequence models, and edit-based systems — we do not see a clear winner. Not surprisingly, we observe that larger models (T5-11B, UL2-20B, Chat-LLaMA-2-7B-FT, Chat-LLaMA-2-13B-FT) have slightly higher Recall compared to smaller models (GECToR-2024, CTC-Copy, EditScorer). This is expressed in 1–2% higher $F_{0.5}$ scores on CoNLLL-2014-test; however, the values on BEA-dev and BEA-test don't show the same behavior.

Additionally, we observe that simply scaling the model does not help achieve a breakthrough in benchmark scores. For example, a relatively small model such as GECToR-2024 ($\approx 300M$ parameters) still performs well enough compared to much larger models ($\approx 7-20B$ parameters). We hypothesize that the limiting factor for English GEC is the amount of high-quality data rather than model size. We have not been able to realize an $F_{0.5}$ score of more than 68% / 59% / 75% on CoNLLL-2014-test / BEA-dev / BEA-test, respectively, with any single-model system approach, which is consistent with previously published results.

For GECToR, after two stages of fine-tuning, we were able to improve the $F_{0.5}$ score of the top-performing single-model model by 3.8% on CoNLL-2014 and by 0.4% on BEA-test, mostly due to the increase in Recall (Table 3).

Interestingly, we see a trend where larger models

[7] https://github.com/AlexeySorokin/EditScorer

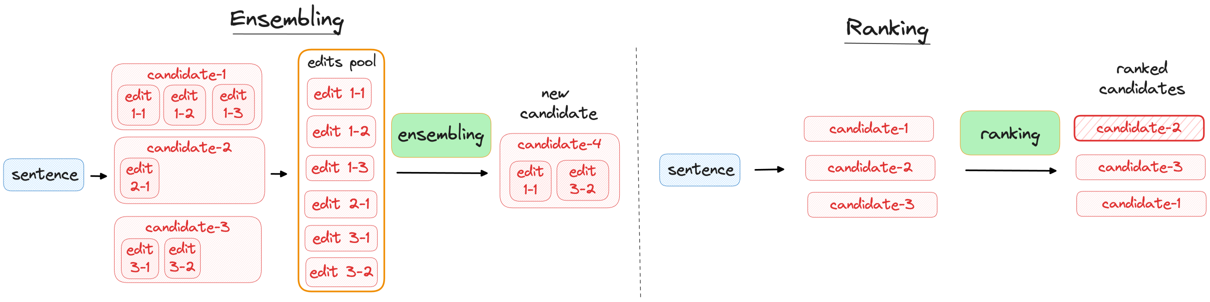


Figure 1: Combining the single-model systems' outputs. Left: In ensembling, candidates (system outputs) are aggregated on an edit level. Right: In ranking, candidates (system outputs) are aggregated on a sentence level. We consider ranking to be a special case of ensembling.

exhibit diminishing returns with multi-staged training approaches. Our exploration of various training data setups reveals that a simple and straightforward approach, focusing exclusively on the W&I + LOCNESS train dataset, performs on par with more complex configurations across both evaluation datasets.

4 Ensembling and Ranking of Single-Model Systems

Combining the outputs of single-model GEC systems can improve their quality. In this paper, we explore two combining methods: ensembling and ranking (Figure 1).

Ensembling combines outputs of single-model systems on an edit level. The ensemble method exploits the strengths of each model, potentially leading to more robust and accurate corrections than any single-model system could provide on its own.

Ranking is a special case of ensembling that combines individual outputs on a sentence level. In this approach, the performance of each system's candidate is assessed against a set of predefined criteria, and the most effective candidate is selected. Ranking maintains the internal coherence of each model's output, potentially leading to more natural and readable corrections.

4.1 Oracle-Ensembling and Oracle-Ranking as Upper-Bound Baselines

To set the upper-bound baseline for our experiments in combining single models, we introduce two *oracle* systems: Oracle-Ensembling and Oracle-Ranking.

Oracle-Ensembling approximates an optimal combination of edits of available single-model systems. It is computationally challenging because the number of possible edit combinations grows exponentially with the number of edits. We use a heuristic to mitigate this; it optimizes Precision at the cost of reducing Recall.

Using golden references from evaluation sets, Oracle-Ensembling works as follows:

1. Aggregate the edits from all systems into a single pool.

2. Identify and select edits that are present in both the edit pool and the available annotation.

3. In the case of multiple annotations, we obtain a set of edits for each annotation separately. We then select the largest set of edits among the multiple annotations.

Oracle-Ranking approximates an optimal output selection for available single-model systems. Again using golden references from evaluation sets, we use M2scorer[8] to obtain $(F_{0.5}, n_{correct}, n_{proposed})$ for each system's output candidate against the available annotation. The output candidates are then sorted by $(+F_{0.5}, +n_{correct}, -n_{proposed})$ and the top one is selected.

For our explorations into combining models' outputs, we select the seven single-model systems that show the best performance on CoNLL-2014-test (Table 2): Chat-LLaMa-2-7B-FT, Chat-LLaMa-2-13B-FT, T5-11B, UL2-20B, GECToR-2024, CTC-Copy, and EditScorer. As our selection criteria, we take i) systems of different types to maximize the diversity and ii) systems that have an $F_{0.5}$ score of at least 65 on CoNLL-2014-test. We refer to this set of models as "best 7".

[8] https://github.com/nusnlp/m2scorer

4.2 Ensembling by Majority Votes on Edit Spans (Unsupervised)

To experiment with ensembling different GEC systems, we needed a method that is tolerant to model architecture and vocabulary size. Ensembling by majority votes (Tarnavskyi et al., 2022) on span-level edits satisfies this requirement, and it's simple to implement, so we decided to start with this approach. We use the same "best 7" set of models in our experiments.

Our majority-vote ensembling implementation consists of the following steps:

0. Initialization. a) Select the set of single-model systems for the ensemble. We denote the number of selected systems by N_{sys}. b) Set N_{min}, the threshold for the minimum number of edit suggestions to be accepted, $0 \leq N_{min} \leq N_{sys}$.

1. Extract all edit suggestions from all single-model systems of the ensemble.

2. For each edit suggestion i, calculate the number of single-model systems n_i that triggered it.

3. Leave only those edit suggestions that are triggered more times than the N_{min} threshold: $\forall i : n_i > N_{min}$.

4. Iteratively apply the filtered edit suggestions, beginning with the edit suggestions with the most agreement across systems (greatest n_i) and ending with the edit suggestions where n_i is lowest. Don't apply an edit suggestion if it overlaps with one of the edits applied on a previous iteration.

4.3 Ensembling and Ranking by GRECO Model (Supervised Quality Estimation)

The quality estimation approach for combining single-model systems' outputs achieved two recent state-of-the-art results: logistic regression-based ESC (Edit-based System Combination) (Qorib et al., 2022), and its evolution, DeBERTA-based GRECO (Grammaticality scorer for re-ranking corrections) (Qorib and Ng, 2023). In this paper, we experiment with GRECO because it is open source and demonstrates state-of-the-art performance on the GEC task to the best of our knowledge[1]. GRECO was trained on the W&I + LOCNESS training set.

We experiment with applying the publicly available GRECO model[9] to the "best 7" set of models. We explore three ways of combining systems' outputs:

GRECO-ens-beam. We reuse beam-search implementation with beam size $k = 16$ on the edit span level.

GRECO-rank. We use GRECO to select the best single-model system's output by choosing the one with the highest score.

GRECO-rank-w. We re-weight GRECO scores for each system's output j by multiplying it by a weighting coefficient w_j:

$$\forall k : w_j = \frac{n_j}{\max(n_k)}, \tag{1}$$

where the numerator n_j is the number of systems that produce this output j, and the denominator $\max(n_k)$ is the maximum number of systems for all systems' outputs. This way, we reduce the score of less frequent systems because it's not the system that is being scored/popular but rather the system's specific output (the edit).

4.4 Ranking by GPT-4 (Zero-Shot)

Besides the direct application of LLMs for GEC in a zero-shot setting (we consider it in the Section 3.1.1), LLMs may be used as a combining method for ensembles. We explore GPT-4 as a ranking tool for single-model GEC systems' outputs.

We use version *gpt-4-0613* for GPT-4 with temperature 1. We implement two prompts, "prompt-a", and "prompt-b", with slightly different goals: prompt-a aims to select the top single-model system's output among the systems' candidates, whereas prompt-b aims to perform the full ranking of the systems' candidates. They both have the same task description. For the following example of ranking three systems, it is:

```
ORIGINAL:
I likes turtles very much.
EDITED:
A: I like turtles very much.
B: I likes turtles very much.
C: I like turtles very much.
```

But they require a different output format:

prompt-a (top cand.): prompt-b (ranking):

```
OUTPUT:                         OUTPUT:
C                               C A B
```

To eliminate potential positional bias, we run each prompt four times with a randomly shuffled order of single-model systems' outputs and average the performance scores. To investigate the impact of the number of systems to be ranked, we evaluate the performance of GPT-4 on two sets of single models: "best 7" and "clust 3".

[9] https://github.com/nusnlp/greco

	CoNLL-2014-test			BEA-dev			BEA-test		
System	Precision	Recall	$F_{0.5}$	Precision	Recall	$F_{0.5}$	Precision	Recall	$F_{0.5}$
ESC (Qorib et al., 2022)	**81.5**	43.8	69.5	**72.9**	**40.4**	62.8	86.6	60.9	79.9
GRECO (Qorib and Ng, 2023), var0*	79.40	48.70	70.48	-	-	**63.4**	86.5	63.1	80.5
GRECO (Qorib and Ng, 2023), var1*	79.60	**49.90**	**71.12**	-	-	-	-	-	-
GRECO (Qorib and Ng, 2023), var2*	-	-	-	-	-	-	**86.7**	**63.7**	**80.8**
Chat-LLaMa-2-13B-FT (single-model system)	**77.3**	45.6	**67.9**	59.8	46.1	56.4	74.6	67.8	73.1
UL2-20B (single-model system)	73.8	**50.4**	67.5	**60.5**	**48.6**	**57.7**	**75.2**	**70.0**	**74.1**
Oracle-Ensembling(best 7), baseline	**100.0**	57.7	**87.2**	**100.0**	58.2	**87.4**	-	-	-
Oracle-Ranking(best 7), baseline	91.4	**64.2**	84.2	79.6	**60.2**	74.7	-	-	-
majority-voting(best 7)	**83.7**	**45.7**	**71.8**	**71.7**	42.2	**62.9**	**87.3**	64.1	**81.4**
majority-voting(best 3)	82.8	44.1	70.4	70.4	**43.1**	62.5	85.1	**64.5**	80.0
GRECO-ens-beam(best 7)	77.3	51.6	70.3	65.5	47.6	60.9	-	-	-
GRECO-rank(best 7)	74.4	**54.2**	69.2	63.2	**50.0**	60.0	-	-	-
GRECO-rank-w(best 7)	**81.6**	49.3	**72.1**	**68.1**	45.8	**62.0**	**82.0**	**67.5**	**78.6**
GPT-4-rank-prompt-a**(clust 3)**	72.4	58.3	69.1	59.7	52.3	58.1	-	-	-
MAJORITY-VOTING ✚[majority-voting(best 7), GRECO-rank-w(best 7)]	83.0	**48.1**	72.5	70.2	**43.9**	62.7	85.6	**65.8**	80.7
MAJORITY-VOTING ✚[majority-voting(best 7), GRECO-rank-w(best 7), GPT-4-rank-a(clust 3)]	**83.9**	47.5	**72.8**	**70.6**	43.5	**62.8**	**86.1**	65.6	**81.1**

"best 7" (best 7 single-model systems): Chat-LLaMa-2-13B-FT + UL2-20B + Chat-LLaMa-2-7B-FT + EditScorer + T5-11B + CTC-Copy + GECToR-2024.
"best 3" (best 3 single-model systems): Chat-LLaMa-2-13B-FT + UL2-20B + Chat-LLaMa-2-7B-FT.
"clust 3" (clustered 3 single-model systems): Chat-LLaMa-2-13B-FT + T5-11B + Edit-Scorer.
*In the paper (Qorib and Ng, 2023), authors prepared different variants of GRECO, each of which is optimized for one test dataset.
**We show mean values across four GPT-4 runs with randomly shuffled single-model systems' outputs.
✚ We denote 2nd order ensembling (ensembles of ensembles) by capital letters.

Table 4: All ensembles evaluated on CoNLL-2014-test, BEA-dev, and BEA-test datasets.

	CoNLL-2014-test			BEA-dev		
	Precision	Recall	$F_{0.5}$	Precision	Recall	$F_{0.5}$
GPT-4-rank-prompt-a(best 7)	70.9 ± 0.5	59.7 ± 0.6	68.4 ± 0.5	56.8 ± 0.3	53.4 ± 0.8	56.1 ± 0.3
GPT-4-rank-prompt-b(best 7)	69.6 ± 0.8	59.5 ± 0.2	67.3 ± 0.7	56.3 ± 0.5	53.9 ± 0.6	55.8 ± 0.4
GPT-4-rank-prompt-a(clust 3)	72.4 ± 0.3	58.3 ± 0.6	69.1 ± 0.1	59.7 ± 0.1	52.3 ± 0.4	58.1 ± 0.1
GPT-4-rank-prompt-b(clust 3)	71.9 ± 0.4	58.1 ± 0.5	68.7 ± 0.5	58.7 ± 0.3	52.0 ± 0.5	57.2 ± 0.3

"best 7" (best 7 single-model systems): Chat-LLaMa-2-13B-FT + UL2-20B + Chat-LLaMa-2-7B-FT + EditScorer + T5-11B + CTC-Copy + GECToR-2024.
"clust 3" (clustered 3 single-model systems): Chat-LLaMa-2-13B-FT + T5-11B + Edit-Scorer.

Table 5: LLM ranking results. We run each prompt four times with randomly shuffled outputs of single-model systems' candidates and report mean ± 2std.

"clust 3" refers to 3 of the 7 best single-model systems: Chat-LLaMa-2-13B-FT + T5-11B + Edit-Scorer. This is the subset of single-model systems from the "best 7" ensemble that provides the most distinct corrections. To select this set, we perform hierarchical clustering on TF-IDF vectors extracted from the BEA-dev dataset using a cosine similarity. The cosine similarity scores are averaged to produce a single matrix that reflects the collective performance of the single-model systems. The dendrogram illustrating the relationships between the systems based on distance is shown in Appendix D, Fig. 2. Based on the threshold $t = 0.11$, we select the three clusters and choose Chat-LLaMa-2-13B-FT, T5-11B, Edit-Scorer to represent each.

4.5 Ensembles of Ensembles

Ensembles may themselves be combined via ensembling or ranking methods to potentially improve performance, and this is an approach we explore as well. We experiment with combining the outputs of three ensemble systems: majority-voting(best 7), GRECO-rank(best 7), and GPT-4-rank(clust 3). Here, majority-voting(best 7) was selected because it achieves the highest $F_{0.5}$ score; GRECO-rank(best 7) and GPT-4-rank(clust 3) have higher Recall and, therefore, potential to add value in an ensemble.

The MAJORITY-VOTING algorithm (we denote second-order ensembling by capital letters) is identical to that described in 4.2.

4.6 Ensembles Results

Oracle ensembling & ranking. Oracle-Ensembling shows $F_{0.5}$ scores of 87.2/87.4 on CoNLL-2014-test/BEA-dev, while Oracle-Ranking performs notably worse with $F_{0.5}$ scores of 84.2/74.7 and Precision of 91.4/79.6 (Table 4). This highlights the high potential for improvements on existing candidate generation and ensembling approaches, whereas ranking is more limited.

Majority-voting ensembling. The only hyperparameter for the method (the N_{min} threshold) directly impacts the Precision/Recall balance: the higher it is set, the greater the Precision. We find that the best N_{min} values for maximizing $F_{0.5}$ score are $N_{min} \approx N_{sys}/2$. **With $\mathbf{N_{min} = 3}$, we achieve 71.8 on CoNLL-2014-test, outperforming the previous state-of-the-art result by 0.7, and 81.4 on BEA-test, setting a new state-of-the-art result.** (Table 4, "best 7" systems ensemble).

We perform an ablation study to measure the impact of each system in the ensemble (Appendix D, Table 12), where we remove systems one by one in the decreasing direction of $F_{0.5}$ score on the BEA-dev dataset. Our experiments show that even an ensemble combined from just the "best 3" systems (Chat-LLaMa-2-13B-FT, UL2-20B, and Chat-LLaMa-2-7B-FT) significantly improves the $F_{0.5}$ score over the UL2-20B single-model system (by 2.9% on CoNLLL-2014-test, 4.8% on BEA-dev, and 5.9% on BEA-test). These results reinforce the significance of ensembling in achieving state-of-the-art performance on the GEC task. We hypothesize that majority-voting ensembling helps in mitigating the influence of noise within the data. By consolidating edits that are consistent across multiple systems (the true signal), and concurrently downplaying less prevalent and potentially inaccurate edits (the noise), the ensembling approach effectively enhances the overall quality and reliability of the output. Our experiments on BEA-dev can be found in Appendix D, Table 8.

Supervised ranking & ensembling. Overall, leveraging GRECO (all variants) for combining systems' outputs leads to increased Recall at the cost of Precision. It leads to an improvement in $F_{0.5}$ score on CoNLLL-2014-test, achieving 72.1% (+0.3% from our best unsupervised ensemble, majority-voting(best 7)). However, results on BEA-test regressed (-2.8% in $F_{0.5}$ score). GRECO-ens-beam did not outperform GRECO-rank-w in our experiments.

Zero-shot ranking. We observe that LLM-based ranking works better for three distinct single-model systems (clust 3) than for all seven best systems (best 7). We hypothesize that this performance disparity may be due to the increased complexity of selecting the optimal choice from a larger set of similar options. We also explain in this way the better performance of prompt-a (selection of the top candidate rewrite) than prompt-b (performing full ranking among candidate rewrites). Similar to GRECO-rank, we notice that GPT-4 favors Recall-oriented outputs, which leads to the highest Recall (58.4) on the CoNLLL-2014-test, but a suboptimal $F_{0.5}$ score. More results are presented in Table 5 and in Appendix D, Table 9.

Ensembles of ensembles. Applying second-order ensembles, more specifically MAJORITY-VOTING[majority-voting(best 7), GRECO-rank-w(best 7), GPT-4-rank-a(clust 3)], helps to even further **push the state-of-the-art record on CoNNL-2014-test, achieving $\mathbf{F_{0.5} = 72.8}$: $+1.7$ compared to the previously highest reported result by GRECO, var1 (Qorib and Ng, 2023)** and $+1.0$ compared to our majority-voting(best 7) ensemble.

5 Related work

Large language models have demonstrated efficacy across a variety of natural language processing tasks, including GEC (Bryant et al., 2023). The comparative analysis conducted by (Wu et al., 2023) on the effectiveness of different models for GEC — ChatGPT, Grammarly, and open-sourced GECToR — reveals that ChatGPT possesses a distinctive capability to enhance textual content by not only correcting errors on a one-by-one basis but also by rephrasing original sentences, changing their structure to maintain grammatical correctness. The outcomes of human evaluations underscore the limitations of exclusively relying on automatic evaluation metrics for assessing GEC model performance, thereby positioning ChatGPT as a potentially invaluable resource for GEC applications.

Other research (Loem et al., 2023), (Fang et al., 2023) suggests that although zero-shot and few-shot chain-of-thought methodologies demonstrate promise in terms of error detection capabilities and the production of fluently corrected text, they generally underperform across the majority of error categories, thus failing to achieve high-quality outcomes in GEC. Moreover, (Zhang et al., 2023c) delved into the customization of open-sourced foundation LLMs including LLaMA (Touvron et al., 2023) for writing assistant applications, with GEC as one of the tasks. The experimental findings indicate that instruction tuning for specific scenarios such as GEC significantly boosts the performance of LLMs and can be used to develop smaller models that outperform their larger, general-purpose counterparts.

Additionally, (Kaneko and Okazaki, 2023) introduced a novel approach for predicting edit spans within source texts, redefining instruction-based fine-tuning as local sequence transduction tasks.

This method not only reduces the length of target sequences but also diminishes the computational demands associated with inference. The study emphasizes that even high-performance LLMs such as ChatGPT struggle to generate accurate edit spans in zero-shot and few-shot scenarios, particularly in the correct generation of indexes, making this approach unstable.

Recent advancements in GEC have largely been attributed to the ensembling of outputs from individual models, as highlighted in studies by (Omelianchuk et al., 2020; Tarnavskyi et al., 2022). When integrating systems with significant disparities, a system combination model is preferred over simple ensembles. This approach allows for effective integration of the strengths of various GEC systems, yielding better results than ensembles, as demonstrated in (Qorib et al., 2022). Model outputs can be re-ranked using majority vote, as well as with the proposed GRECO model (Qorib and Ng, 2023), a new state-of-the-art quality estimation model correlating more closely with the $F_{0.5}$ score of a corrected sentence, thus leading to a combined GEC system with a higher $F_{0.5}$ score. Additionally, this study proposes three methods for leveraging GEC quality estimation models in system combination: model-agnostic, model-agnostic with voting bias, and model-dependent methods.

Conclusions

We don't find that any single-model system approach is dominant across all benchmarks. While in general, fine-tuning the larger models leads to higher $F_{0.5}$ scores, the 10–50x increase in model size leads to rather small improvements (up to 1–2 $F_{0.5}$ points). We hypothesize that the main bottleneck in improvement is high-quality data rather than system's architecture or model size.

To date, ensembling is crucial to overcome the limitations of single-model system approaches. Even a simple heuristic approach such as majority voting with just three single-model systems significantly boosts the quality (by 3–6 $F_{0.5}$ points). While more complex approaches (supervised ensembling or LLM zero-shot ranking) may lead to potentially better results (more specifically, show higher Recall), they usually do not lead to the target metric: $F_{0.5}$ improvement on GEC benchmarks.

Recent LLM-powered methods do not outperform other available approaches to date. However, being properly set, they can perform on par with other methods and lead to more powerful ensembles.

We've not yet reached the ceiling on the existing GEC benchmarks. Our research shows that it's possible to improve previous records noticeably, setting the new state-of-the-art performance on two principal GEC benchmarks with $F_{0.5}$ scores of 72.8 on CoNLL-2014-test and 81.4 on BEA-test, which are improvements of +1.7 and +0.6, respectively.

In future work, we plan to explore the generation of high-quality synthetic GEC data powered by a state-of-the-art ensemble. We hypothesize that this could democratize the field by reducing the necessity of expensive training of large models to achieve a superior level of quality.

6 Acknowledgements

We express our gratitude to our colleagues Viacheslav Klimkov, Max Gubin, Viktor Zamaruiev for their valuable advices and support, and to Paige Schwartz for careful editing. We would like to thank the reviewers for their thoughtful comments and efforts towards improving our paper. To our communities: While we are writing this, our homeland Ukraine continues to resist the unprovoked Russian invasion. We are grateful to everyone who defends Ukraine, declares support to the people of Ukraine, and is sending aid. Thank you!

Limitations

Firstly, our analysis was confined to the English language, potentially limiting the generalizability of our findings to other languages with potentially different error correction challenges.

Next, our evaluation relied on two specific benchmarks using automated metrics, without incorporating human evaluation to assess the quality of the GEC. While automated metrics provide a scalable and objective means of evaluation, they may not fully capture the nuances of language that human judgment can offer.

Additionally, as we focus on ensembles, our research does not address the speed performance of the proposed systems. Therefore our findings may not provide a comprehensive view of the practicality and scalability of the proposed methods.

Lastly, the use of closed-source proprietary LLMs introduces a layer of uncertainty, as these models may undergo changes over time that are not publicly disclosed. Such changes could potentially affect the reproducibility of our results.

References

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.

Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. 2023. Grammatical error correction: A survey of the state of the art. *Computational Linguistics*, page 1–59.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2014. One billion word benchmark for measuring progress in statistical language modeling.

Steven Coyne, Keisuke Sakaguchi, Diana Galvan-Sosa, Michael Zock, and Kentaro Inui. 2023. Analyzing the performance of gpt-3.5 and gpt-4 in grammatical error correction. *ArXiv*, abs/2303.14342.

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.

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*.

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.

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.

Masahiro Kaneko and Naoaki Okazaki. 2023. Reducing sequence length by predicting edit operations with large language models. *ArXiv*, abs/2305.11862.

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.

Mengsay Loem, Masahiro Kaneko, Sho Takase, and Naoaki Okazaki. 2023. Exploring effectiveness of GPT-3 in grammatical error correction: A study on performance and controllability in prompt-based methods. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 205–219, Toronto, Canada. 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.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 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.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

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.

Muhammad Reza 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, Seattle, United States. Association for Computational Linguistics.

Muhammad Reza Qorib and Hwee Tou Ng. 2023. System combination via quality estimation for grammatical error correction. In *Proceedings of the 2023*

Conference on Empirical Methods in Natural Language Processing, pages 12746–12759, Singapore. 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. *J. Mach. Learn. Res.*, 21(1).

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.

Jonathan Schler, Moshe Koppel, Shlomo Engelson Argamon, and James W. Pennebaker. 2006. Effects of age and gender on blogging. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*.

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.

Raymond Hendy Susanto, Peter Phandi, and Hwee Tou Ng. 2014. System combination for grammatical error correction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 951–962, Doha, Qatar. 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.

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.

Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Neil Houlsby, and Donald Metzler. 2022. Unifying language learning paradigms. *arXiv preprint arXiv:2205.05131*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

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

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.

Yu Zhang, Yue Zhang, Leyang Cui, and Guohong Fu. 2023a. Non-autoregressive text editing with copy-aware latent alignments. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7075–7085, Singapore. Association for Computational Linguistics.

Yue Zhang, Leyang Cui, Deng Cai, Xinting Huang, Tao Fang, and Wei Bi. 2023b. Multi-task instruction tuning of llama for specific scenarios: A preliminary study on writing assistance.

Zhihan Zhang, Wenhao Yu, Mengxia Yu, Zhichun Guo, and Meng Jiang. 2023c. A survey of multi-task learning in natural language processing: Regarding task relatedness and training methods. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 943–956, Dubrovnik, Croatia. Association for Computational Linguistics.

A Hierarchical clustering analysis for single-model systems

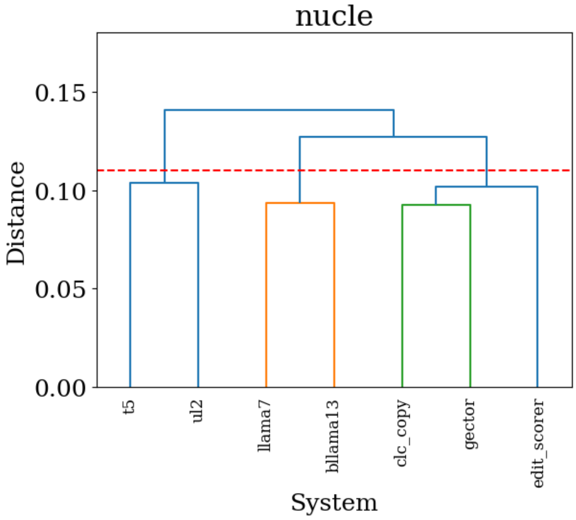


Figure 2: Dendrogram of hierarchical clustering analysis for single-model systems. The y-axis represents the distance metric used for clustering, with a red dashed line indicating the selected threshold for cluster formation ($t = 0.11$). The x-axis enumerates different systems that were analyzed. The dendrogram branches reflect the hierarchical grouping based on the proximity of distance metrics.

B Unsuccessful attempt to reproduce Seq2Edit approach

The sequence-to-edit approach leverages the fact that in GEC, the target sentence is usually very similar to the source one. Instead of rewriting the entire sentence, it's possible to generate a list of required edits, represented as tuples: (start position, end position, replacement). (Stahlberg and Kumar, 2020). We tried to re-implement the most recent approach (Kaneko and Okazaki, 2023) that reported a high score ($F_{0.5} = 71.3\%$) on the CoNLL-2014-test. We attempted to fine-tune both T5-11B and LLaMA-2-7B models using the same set of hyperparameters that we used in our other experiments, on pairs of sentences and edits extracted from the BEA-train dataset. We were unable to get any meaningful results (our $F_{0.5}$ on CoNLL-2014-test was about 30, which is around 40 points lower than SOTA systems). Our models tended to corrupt an original sentence more often than correct it. We believe that our implementation most likely misses some crucial details required to work properly, and we encourage other researchers to reproduce and open-source the sequence-to-edit approach.

C Second-order ensembling of LLM-containing ensembles by aggressiveness ranking

AGGR-RANK is a ranking method that takes as input two ensembles: GPT-4-rank and an alternative ensemble. It selects GPT-4-rank under two conditions: 1) it is less "aggressive" than the alternative (it suggests fewer edited spans), and 2) it is non-trivial (edits do exist).

The results are presented in Table 10. The first system (AGGR-RANK✚[GPT-4-rank-a(clust 3), majority-voting(best 7)]) tends to have a higher Precision across all datasets. The second system (AGGR-RANK✚[GPT-4-rank-a(clust 3), GRECO-rank-w(best 7)]), despite its lower Precision, manages to achieve a slightly higher $F_{0.5}$ score on the CoNLL-2014 test dataset, suggesting that its improved Recall adequately compensates in this case. Overall, the F_0.5 score is generally higher for the first system on CoNLL-2014 test and BEA-test, indicating that second-order ensembling on top of the GRECO approach is the most favorable.

D Ablation studies

Model	**Datasets used for training**			**CoNLL-2014-test**			**BEA-dev**		
	NUCLE	**W&I**	**cLang-8**	**Precision**	**Recall**	**$F_{0.5}$**	**Precision**	**Recall**	**$F_{0.5}$**
LLaMA-2-7B-FT	-	ALL	-	68.66	**54.27**	65.20	57.90	**48.63**	55.77
LLaMA-2-7B-FT	-	-	ALL	67.25	50.44	63.05	57.99	42.11	53.93
LLaMA-2-7B-FT	ALL	ALL	-	72.45	46.98	65.37	58.00	45.82	55.07
Chat-LLaMa-2-7B-FT	ALL	-	-	70.39	36.31	59.42	50.72	24.51	41.79
Chat-LLaMa-2-7B-FT	-	ALL	-	70.45	52.59	65.97	**59.19**	47.81	**56.50**
Chat-LLaMa-2-7B-FT	-	ALL	100k	68.94	52.78	64.96	57.94	45.53	54.94
Chat-LLaMa-2-7B-FT	ALL	ALL	48k	**75.40**	46.84	**67.20**	58.26	46.03	55.32
Chat-LLaMa-2-7B-FT	TP, 8k	TP, 8k	TP, 24k	68.01	52.84	64.32	53.94	46.03	52.15
Chat-LLaMa-2-13B-FT	ALL	ALL	100k	**77.34**	**45.57**	**67.87**	**59.79**	**46.08**	**56.43**

Table 6: A search of the best dataset combination for fine-tuning large language models. For fine-tuned models, different training dataset combinations were evaluated: Here, "ALL" denotes the usage of all available data for training, specific numbers (e.g., "100k") define the specific number of samples used for training, and "TP" ("true positives") denotes when only the dataset's samples containing corrections are used.

Hyperparameter	**Values for stage I**	**Values for stage II**
train data source	cLang8, BEA-train, 20 Troy	BEA-train
train data size	2,897,676	33,618
batch_size	8	16
accumulation_size	32	16
n_epoch	5	4
patience	3	3
max_len	50	50
LR	1e-05	1e-05
cold_steps_count	0	0
tp_prob	1	1
tn_prob	1	1
updates_per_epoch	10000	0
special_tokens_fix	1	1
transformer_model	Roberta-large	Roberta-large
Pretrained model	roberta-large_1_pie_1bw_st3	roberta-stage1
Inference tweaks: minimum error probability	0.65	0.65
Inference tweaks: confidence	0.1	0.1

Table 7: Hyperparameter values for the fine-tuning of GECToR-2024.

		BEA-dev		
System name	$\mathbf{N_{min}}$	**Precision**	**Recall**	$\mathbf{cF_{0.5}}$
majority-voting(best 7)	3	71.7	42.2	62.9
majority-voting(best 7) w/o GECToR-2024	3	73.8	39.1	**62.7**
majority-voting(best 7) w/o CTC-copy	3	73.7	39.0	62.6
majority-voting(best 7) w/o EditScorer	3	72.8	39.5	62.3
majority-voting(best 7) w/o T5-11B	3	74.2	35.8	61.1
majority-voting(best 7) w/o UL2-20B	3	74.2	35.9	61.1
majority-voting(best 7) w/o LlaAMA-2-7B	3	74.3	36.2	61.4
majority-voting(best 7) w/o LlaAMA-2-13B	3	74.3	36.2	61.3
majority-voting(best 6) (best 7 w/o GECToR)	3	73.8	39.1	62.7
majority-voting(best 6) w/o CTC-copy	2	69.8	44.5	62.7
majority-voting(best 6) w/o EditScorer	2	69.0	45.3	62.5
majority-voting(best 6) w/o T5-11B	2	70.6	42.4	62.3
majority-voting(best 6) w/o UL2-20B	2	70.6	42.5	62.3
majority-voting(best 6) w/o Llama-2-7B	2	71.5	43.2	**63.2**
majority-voting(best 6) w/o Llama-2-13B	2	71.1	43.1	63.0
majority-voting(best 5) (best 6 w/o Llama-2-7B)	2	71.5	43.2	63.2
majority-voting(best 5) w/o CTC-copy	2	74.0	38.8	**62.6**
majority-voting(best 5) w/o EditScorer	2	72.6	39.2	62.0
majority-voting(best 5) w/o T5-11B	2	75.1	33.8	60.3
majority-voting(best 5) w/o UL2-20B	2	74.8	34.0	60.3
majority-voting(best 5) w/o LlaMA-2-13B	2	74.7	34.9	60.8
majority-voting(best 4) (best 5 w/o CTC-copy)	2	74.0	38.8	62.6
majority-voting(best 4) w/o EditScorer	1	66.2	47.9	61.5
majority-voting(best 4) w/o T5-11B	1	70.4	43.1	**62.5**
majority-voting(best 4) w/o UL2-20B	1	69.9	43.7	62.4
majority-voting(best 4) w/o LlaMA-2-13B	1	68.5	45.2	62.1
majority-voting(best 3) (best 4 w/o T5-11B)	1	70.4	43.1	62.5
majority-voting(best 3) w/o EditScorer	1	72.9	36.4	**60.7**
majority-voting(best 3) w/o UL2-20B	1	77.0	28.0	57.0
majority-voting(best 3) w/o LlaMA-2-13B	1	77.3	29.2	58.2

Table 8: Ablation study of removing single-model GEC systems from majority-based ensembles on BEA-dev.

"best 7" (best 7 single-model systems): Chat-LLaMa-2-13B-FT + UL2-20B + Chat-LLaMa-2-7B-FT + EditScorer + T5-11B + CTC-Copy + GECToR-2024.
"best 6" (best 6 single-model systems): Chat-LLaMa-2-13B-FT + UL2-20B + Chat-LLaMa-2-7B-FT + EditScorer + T5-11B + CTC-Copy.
"best 5" (best 5 single-model systems): Chat-LLaMa-2-13B-FT + UL2-20B + Chat-LLaMa-2-7B-FT + EditScorer + T5-11B.
"best 4" (best 4 single-model systems): Chat-LLaMa-2-13B-FT + UL2-20B + Chat-LLaMa-2-7B-FT + EditScorer.
"best 3" (best 3 single-model systems): Chat-LLaMa-2-13B-FT + UL2-20B + Chat-LLaMa-2-7B-FT.

	CoNLL-2014-test			BEA-dev		
	Precision	**Recall**	**$F_{0.5}$**	**Precision**	**Recall**	**$F_{0.5}$**
Chat-LLaMa-2-13B-FT	77.3	45.6	67.9	59.8	46.1	56.4
T5-11B	70.9	56.5	67.5	60.9	51.1	58.6
GPT-4-rank-a(best 7)	71.2	60.1	68.7	56.9	53.8	56.2
	71.0	59.5	68.4	56.9	53.1	56.1
	70.7	59.5	68.2	56.6	53.1	55.9
	70.7	59.8	68.2	56.8	53.7	56.2
mean ± 2std	70.9 ± 0.5	59.7 ± 0.6	68.4 ± 0.5	56.8 ± 0.3	53.4 ± 0.8	56.1 ± 0.3
GPT-4-rank-b(best 7)	69.2	59.6	67.0	56.2	53.8	55.7
	69.6	59.5	67.3	56.0	53.5	55.5
	69.5	59.4	67.2	56.6	54.0	56.0
	70.2	59.6	67.8	56.3	54.2	55.9
mean ± 2std	69.6 ± 0.8	59.5 ± 0.2	67.3 ± 0.7	56.3 ± 0.5	53.9 ± 0.6	55.8 ± 0.4
GPT-4-rank-a(clust 3)	72.3	58.4	69.0	59.8	52.2	58.1
	72.2	58.6	69.0	59.7	52.5	58.1
	72.6	57.9	69.1	59.7	52.1	58.0
	72.4	58.4	69.1	59.7	52.5	58.1
mean ± 2std	72.4 ± 0.3	58.3 ± 0.6	69.1 ± 0.1	59.7 ± 0.1	52.3 ± 0.4	58.1 ± 0.1
GPT-4-rank-b(clust 3)	71.7	57.8	68.4	58.7	51.7	57.2
	71.8	58.2	68.6	58.5	51.8	57.0
	72.2	58.4	69.0	58.9	52.1	57.4
	71.9	58.1	68.7	58.7	52.2	57.2
mean ± 2std	71.9 ± 0.4	58.1 ± 0.5	68.7 ± 0.5	58.7 ± 0.3	52.0 ± 0.5	57.2 ± 0.3

Table 9: LLM ranking for "best 7" (best 7 single-model systems): Chat-LLaMa-2-13B-FT + UL2-20B+ Chat-LLaMa-2-7B-FT + EditScorer + T5-11B + CTC-Copy + GECToR-2024) and "clust 3" (clustered 3 single-model systems: Chat-LLaMa-2-13B-FT + T5-11B + Edit-Scorer). We denote "prompt-a" (top candidate) as "GPT-4-rank-a", and "prompt-b" (ranking candidates) as "GPT-4-rank-b". We run each prompt four times with randomly shuffled outputs of single-model systems' candidates.

	CoNLL-2014-test			BEA-dev			BEA-test		
System	**Precision**	**Recall**	**$F_{0.5}$**	**Precision**	**Recall**	**$F_{0.5}$**	**Precision**	**Recall**	**$F_{0.5}$**
AGGR-RANK ✚[GPT-4-rank-a(clust 3), majority-voting(best 7)]	**84.0**	45.4	71.8	**71.7**	41.7	62.7	**87.5**	63.8	**81.4**
AGGR-RANK ✚[GPT-4-rank-a(clust 3), GRECO-rank-w(best 7)]	81.9	**49.0**	72.2	68.3	**45.1**	61.9	82.4	**67.0**	78.8

"best 7" (best 7 single-model systems): Chat-LLaMa-2-13B-FT + UL2-20B + Chat-LLaMa-2-7B-FT + EditScorer + T5-11B + CTC-Copy + GECToR-2024.
"clust 3" (clustered 3 single-model systems): Chat-LLaMa-2-13B-FT + T5-11B + Edit-Scorer.
*In the paper (Qorib and Ng, 2023), authors prepared different variants of GRECO, each of which is optimized for one test dataset.
**We show mean values across four GPT-4 runs with randomly shuffled single-model systems' outputs.
✚ We denote 2nd order ensembling (ensembles of ensembles) by capital letters.

Table 10: Second-order ensembling by aggressiveness ranking.

Model	**Instructions**	**CoNLL-2014-test**			**BEA-dev**		
	are used	**Precision**	**Recall**	**$F_{0.5}$**	**Precision**	**Recall**	**$F_{0.5}$**
LLaMA-2-7B-FT	No	**69.33**	50.26	64.44	**59.45**	46.29	**56.25**
LLaMA-2-7B-FT	Yes	68.66	**54.27**	**65.20**	57.9	**48.63**	55.77
Chat-LLaMa-2-7B-FT	No	67.53	**53.59**	64.19	58.00	47.37	55.51
Chat-LLaMa-2-7B-FT	Yes	**70.45**	52.59	**65.97**	**59.19**	**47.81**	**56.50**
LLaMA-2-7B-FT	Yes	68.66	54.27	65.20	57.9	48.63	55.77
LLaMA-2-13B-FT	Yes	**71.49**	**55.67**	**67.65**	**60.28**	**49.26**	**57.69**
Chat-LLaMa-2-7B-FT	Yes	70.45	52.59	65.97	**59.19**	47.81	56.50
Chat-LLaMa-2-13B-FT	Yes	**72.35**	**54.48**	**67.90**	59.04	**48.73**	**56.64**

Table 11: Ablation study on instructions' usage in fine-tuned on W&I dataset Large Language Models.

	CoNLL-2014-test			BEA-dev			BEA-test		
System	**Precision**	**Recall**	$\mathbf{F_{0.5}}$	**Precision**	**Recall**	$\mathbf{F_{0.5}}$	**Precision**	**Recall**	$\mathbf{F_{0.5}}$
majority-voting(best 7), $N_{min} = 3$	83.7	45.7	**71.8**	71.7	42.2	62.9	87.3	64.1	**81.4**
majority-voting(best 6), $N_{min} = 3$	85.3	41.7	70.5	73.8	39.1	62.7	89.0	60.6	81.4
majority-voting(best 5), $N_{min} = 2$	83.0	**46.3**	71.7	71.5	**43.2**	**63.2**	86.4	**64.7**	81.0
majority-voting(best 4), $N_{min} = 2$	86.4	40.4	70.3	**74.0**	38.8	62.6	**88.8**	59.9	81.0
majority-voting(best 3), $N_{min} = 1$	82.8	44.1	70.4	70.4	43.1	62.5	85.1	64.5	80.0
majority-voting(best 2), $N_{min} = 1$	**86.9**	36.3	67.9	72.9	36.4	60.7	86.9	57.8	78.9

"best 7" (best 7 single-model systems): Chat-LLaMa-2-13B-FT + UL2-20B + Chat-LLaMa-2-7B-FT + EditScorer + T5-11B + CTC-Copy + GECToR-2024.
"best 6" (best 6 single-model systems): Chat-LLaMa-2-13B-FT + UL2-20B + Chat-LLaMa-2-7B-FT + EditScorer + T5-11B + CTC-Copy.
"best 5" (best 5 single-model systems): Chat-LLaMa-2-13B-FT + UL2-20B + Chat-LLaMa-2-7B-FT + EditScorer + T5-11B.
"best 4" (best 4 single-model systems): Chat-LLaMa-2-13B-FT + UL2-20B + Chat-LLaMa-2-7B-FT + EditScorer.
"best 3" (best 3 single-model systems): Chat-LLaMa-2-13B-FT + UL2-20B + Chat-LLaMa-2-7B-FT.
"best 2" (best 2 single-model systems): Chat-LLaMa-2-13B-FT + UL2-20B.

Table 12: Ablation study for majority-voting ensembles.

Using Adaptive Empathetic Responses for Teaching English

Li Siyan, Teresa Shao, Zhou Yu, Julia Hirschberg
Department of Computer Science
Columbia University
{siyan.li,ts3488,zy2461,jbh2019}@columbia.edu

Abstract

Existing English-teaching chatbots rarely incorporate empathy explicitly in their feedback, but empathetic feedback could help keep students engaged and reduce learner anxiety. Toward this end, we propose the task of negative emotion detection via audio, for recognizing empathetic feedback opportunities in language learning. We then build the first spoken English-teaching chatbot with adaptive, empathetic feedback. This feedback is synthesized through automatic prompt optimization of ChatGPT and is evaluated with English learners. We demonstrate the effectiveness of our system through a preliminary user study.

1 Introduction

Teacher empathy has been shown to improve the learning experience, including increasing learner engagement and reducing anxiety (Cooper, 2002; Lam et al., 2011; Zhang, 2022b). Recently, Wu et al. (2023) suggests that students' perceived affective support (PAS) from teachers has a positive correlation with *L2 grit*, defined as the passion and perseverance for second-language learning (Teimouri et al., 2022). PAS generally corresponds to the perceived level of support for emotional needs. Its definition includes caring, valuing responses, listening, and encouragement (Sakiz, 2007). We therefore expect empathy to correlate positively with PAS. We aim to examine whether an empathetic, English-teaching system with high PAS similarly boosts L2 grit.

English-teaching systems have adopted affective components for various purposes (Zhai and Wibowo, 2022). However, the systems that have introduced empathetic components into pedagogy are either situated in ubiquitous learning environments (Dai et al., 2014; Santos et al., 2016) or are not naturalistic or seamless in their approaches to accounting for student affect (Wu et al., 2022). An interactive system that effectively detects and alleviates ESL learner anxiety without sensors (e.g. pulse rate monitors) or cameras has yet to be implemented.

Detecting negative emotion from a learner's audio is a promising way to offer empathetic feedback. However, off-the-shelf English speech emotion recognition models are often trained on data collected with native speakers of English (Busso et al., 2008; Lotfian and Busso, 2017). We hypothesize that English spoken by non-native speakers will have differences (Lin, 2014) that challenge these off-the-shelf models. To address this, we develop a preliminary pipeline for this task using annotated audio data and incorporate it into a spoken empathetic chatbot system.

Our spoken chatbot detects negative emotions or prolonged pauses and then responds empathetically to encourage students. This negative-emotion-responsive design is inspired by an automated physics tutor that senses student frustration using sensors and cameras (D'mello and Graesser, 2013). We currently employ model-based and automatic approaches for sensing negative affect in user audio. The chatbot also provides grammar feedback. We utilize a grammar correction model for grammatical feedback and ChatGPT with optimized prompting for empathetic feedback. Positive preliminary user study results indicate that users perceive affective support from our system, paving the way for future large-scale experiments to study our system's effect on learner L2 grit.

Our main contributions are: 1) We release a dataset of Mandarin-accented English speech with high-quality ASR transcripts and negative emotion annotations, and 2) We propose the first sensor-free educational English chatbot that detects negative affect and intervenes by providing adaptive empa-

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 34–53
June 20, 2024 ©2024 Association for Computational Linguistics

thetic feedback [1].

2 Related Work

2.1 Emotion Recognition in English-Teaching Systems

Past English-teaching systems often relied on facial emotion recognition for detecting user affective states. Lin et al. (2015) features a teaching assistant that recognizes the user's emotional state from facial expressions and then adjusts the material's difficulty. Zhang (2022a) proposes a convolutional neural network-based approach to learner emotion recognition to be used in future systems. We are not considering the visual modality due to the constraints of the dialogue framework we build upon.

Mazur et al. (2011) creates a gamified scoring system to adapt to different users. This system is equipped with affect classification for Japanese textual input, yet the role of empathy here is unclear.

Other systems have employed less seamless approaches to detecting affect changes. Wu et al. (2022) constructs an emotion recognition module by recording the number of times a learner clicks on positive and negative emoticons. Santos et al. (2016) employs Arduino, an open-source electronic prototyping platform to detect learner physiological changes, such as pulses and skin conductivity. Another ubiquitous learning system, Dai et al. (2014), uses speech signal and multi-agent behavioral data for online learning and a neural mechanism model for analyzing learners' emotional characteristics.

2.2 Affective English-Teaching Chatbots

Chatbots are effective in increasing student conversational activity during discussions (Goda et al., 2014), improving listening skills (Kim, 2018) and grammar (Kim et al., 2019), and enhancing writing abilities (Lin and Chang, 2020). Since ChatGPT appeared, the quality of chatbot responses has improved dramatically, eliminating concerns about adverse effects on student outcomes due to low response quality (Fryer et al., 2020).

Ayedoun et al. (2015) introduces a multimodal agent that simulates a restaurant waiter to situate participants in a social conversational context to improve their willingness to communicate. Ayedoun et al. (2020) further improves this system by incorporating communication strategies and affective backchannels to provide personalized scaffolding. While the systems alleviate learner anxiety, learner emotions are not directly accounted for or addressed. Both systems also rely on pre-scripted dialogue and are restricted in scenarios.

Shi et al. (2020) builds an empathetic spoken chatbot into a WeChat program for English tutoring. The GPT-2-based (Radford et al., 2019) chatbot utilizes an ontology and a retrieval-based generation approach similar to XiaoIce (Zhou et al., 2020). Despite being empathetic, the bot only uses audio for pronunciation correction.

2.3 Pauses and Anxiety in ESL Context

Foreign language anxiety can correlate with higher pause rates and lower fluency. Pérez Castillejo (2019) established that learners with higher language anxiety tend to pause more frequently. In a study by Wilang and Vo (2018) that monitors ESL speakers speaking during an exam, pausing is associated with heart rate spikes for some, indicating anxiety during pauses. ESL teachers have also noted pauses and stammering as signs of students struggling with language anxiety (Kasap, 2019).

3 System Design

3.1 Overview

Figure 1 shows a system overview: User audio is sent to the *Empathetic Feedback* module to determine whether the user is distressed. If so, the bot produces empathetic feedback using past user utterances; otherwise, the system continues to the *Grammatical Feedback* stage, where grammar critiques are given if applicable. If either feedback mechanism is triggered, the system transitions back to the original conversation through the *User Query Response* stage if the user follows up with the feedback, then through the *Connect Feedback & Conversation* module. To avoid overwhelming users, we ensure at least two turns between grammatical feedback and four turns between empathetic feedback. We discussed our design with ESL students and consulted teachers before finalizing our system. See Appendix A for details.

We build on an existing dialogue framework (Li et al., 2022) for speech and text dialogue system development. The system allows users to converse with the chatbot by recording their utterances through a microphone. The utterance is then converted to text using Whisper medium (Radford et al., 2023) and the text and audio are sent to the chatbot for further analyses and response synthesis.

[1] The dataset and code are in `https://github.com/siyan-sylvia-li/adaptive_empathetic_BEA2024`

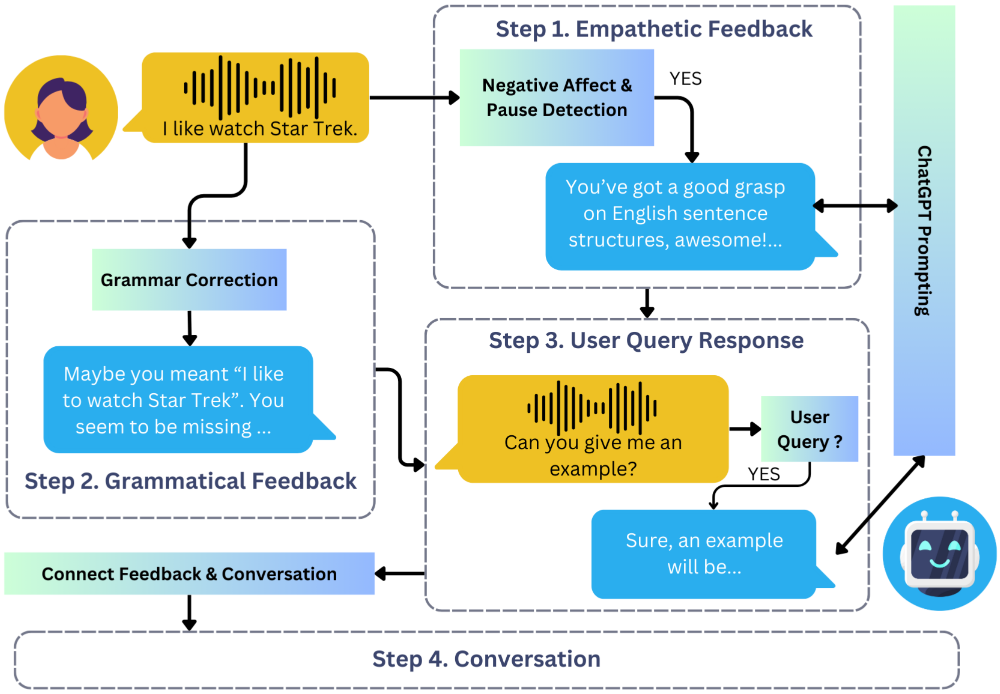

Step 1. Empathetic Feedback
I like watch Star Trek.
Negative Affect & Pause Detection
YES
You've got a good grasp on English sentence structures, awesome!...
ChatGPT Prompting
Grammar Correction
Maybe you meant "I like to watch Star Trek". You seem to be missing ...
Step 2. Grammatical Feedback
Step 3. User Query Response
Can you give me an example?
User Query ?
YES
Sure, an example will be...
Connect Feedback & Conversation
Step 4. Conversation

Figure 1: System Design Overview.

The chatbot response is spoken using SpeechT5 (Ao et al., 2022). For the specific speaker embedding, we selected one of the slt clips from the CMU Arctic speech databases(Kominek and Black, 2004) manually. When choosing the speaker embedding, we aimed for a female voice that can sufficiently induce perceived empathy.

3.2 Data Used

To create data for testing various modules, we utilized audio clips of native Mandarin speakers conversing with a chatbot collected from an English practice platform (Li et al., 2022). 3,200 audio clips from 613 conversations and 163 users remained after filtering. The filtering process removes audio clips containing only Mandarin, duplicates, and a subset of self-introductions from the users. We were not able to eliminate all identifying information from this stage of filtering, but we will remove all identifiable information before publicizing our data. Each audio clip ranges from one second to two minutes. We did not collect demographic information for user identity protection.

We transcribed all audio clips with Whisper medium for training the text-based models in our pipeline. Whisper is not always sufficiently robust to handle heavily accented speech in our data; however, to realistically simulate the environment for our models, we choose not to correct these transcriptions, although we will release the data after manually correcting the transcripts to ensure quality. Realistically, our system should improve as more accent-robust real-time ASR systems emerge.

3.3 Grammatical Feedback

Grammar Correction Model: Following the framework in Liang et al. (2023), we train a grammar correction model to modify user utterance transcripts. We originally prompted ChatGPT for grammar correction feedback. However, responses were often hallucinated or malformed, including using the original utterance as the correction despite correctly identifying grammatical mistakes. Therefore, we train a Llama-2-7b (Touvron et al., 2023) model on ErAConD (Yuan et al., 2022), which contains high-quality error-correction pairs collected from human-chatbot written dialogues. Since grammar correction is a sequence-to-sequence task, we train additional Flan-T5 models (Chung et al., 2022) on the same data. We include more details about the training process in Appendix D.

To evaluate the models, we compute the exact match scores between model predictions and the ground truth corrections in the ErAConD test set. Llama occasionally extends its output (See Table 1), so we include another criterion, substring match, to indicate whether the ground truth is included in

Input	Correction	Llama
I like to read book and study English.	I like to read books and study English	I like to read books and study English. I also like to spend time with my friends.
Love story	Love story.	Love story. Maybe I will write a book one of these days.

Table 1: Examples of the trained Llama model extending the original output.

Model	EM	SM	Corr.
Flan-T5-base	0.56	0.65	N/A
Flan-T5-XL	0.6	0.68	0.53
Flan-T5-XXL	**0.62**	**0.72**	**0.58**
Llama-2-chat-7b	0.30	0.68	**0.58**

Table 2: Exact match scores, substring match scores, and GPT correction scores for different grammar correction models.

the prediction. We also evaluate grammar correction quality on transcribed spoken utterances for Flan-T5-XL, XXL, and Llama. Our trained models correct 100 transcribed spoken utterances. Due to the lack of ground truth grammar corrections, we use AI feedback from GPT-4-Turbo to assess if each prediction is grammatically correct. The results of the evaluation are shown in Table 2.

We observe an increase from exact match to substring match across the board because the ground truth grammatical corrections do not always append periods, while most trained models do. As we transition to out-of-domain data (from written to transcription), we see a decrease in correction accuracy. However, this drop is the smallest for Llama, suggesting higher generalizability to out-of-domain data. We, therefore, choose Llama for our grammar correction model for its relatively higher robustness and smaller size than Flan-T5-XXL.

Grammatical Feedback Format: We would like to present grammar model corrections to the students. Upon considering our design survey results, we choose conversational recasts (Lyster et al., 2013). This involves reformulating student utterances, often including confirmation checks (e.g. "Did you mean [corrected sentence]?"). We implement the recast by pre-pending the corrected sentence with a random confirmation check phrase (e.g. "I think you meant"). When the corrected sentence is longer than 20 words, we instead identify a dependency parse constituent containing the error to avoid repeating the entire sentence when possible. Since the sentences are sentence-tokenized before being corrected, we ignore Llama corrections longer than one sentence. This addresses the previous Llama extension issue.

In addition to a conversational recast, we want to explain how the student's utterance is incorrect. We utilize the conversational grammar correction feedback templates proposed in Liang et al. (2023) and append the templated feedback to the utterance. See examples of our grammatical feedback in Appendix G.

3.4 Negative Emotion and Pause Detection

Data Labeling: Since no accented speech emotion classification dataset exists, we labeled our audio clips to create evaluation data for our pipeline. We used four labels: Negative, Pauses, Neutral, and Unusable. Two Mandarin native speakers with high English proficiency annotated approximately 10% of the data with a Kappa of 0.893. We only include audio clips whose labels both annotators agreed upon. Our audio dataset's data distribution and label definitions are in Table 3. The label definitions were presented to the annotators as the annotation scheme. The annotators also labeled clips featuring both negative affect and pauses as "Negative" to promote better label balance, since students rarely display negative emotions in our data.

Negative Emotion Detection: Because of the shortage of emotion-labeled accented speech data, we could not train new audio classification models for our specific task. Instead, we manipulate a popular out-of-the-box speech emotion classification model[2]. We test different configurations and settings for this model on the small emotion-labeled dataset from the previous segment. Specifically, given the output probabilities for different emo-

[2] https://huggingface.co/ehcalabres/wav2vec2-lg-xlsr-en-speech-emotion-recognition

Label	Definition	Counts
Unusable	The audio is either completely silent, the speaker(s) are conversing in Mandarin, or the utterance is completely unintelligible.	8
Negative	The speaker displays negative sentiments: e.g. anger, frustration, or sadness. Include instances in which the speaker displays an unwillingness to communicate. Include instances where the speaker asks for clarification, as it is an implicit display of confusion.	39
Pauses	The speaker makes many pauses during their utterance. These pauses make it sound like the speaker is struggling to construct the sentences.	54
Neutral	This includes all usable clips that are labeled neither negative nor pauses.	200

Table 3: Definitions for different labels in our data labeling process and their corresponding counts in our labeled audio dataset. These label definitions were presented to the annotators.

Metric	Clip Label	Mean	Diff
Ratio	Pauses	0.41	
	Neutral	0.32	0.09
Pause Rate	Pauses	0.60	
	Neutral	0.55	0.05
Pause Length	Pauses	0.68	
	Neutral	0.49	**0.19**

Table 4: The three metrics for the clips labeled as "Pauses" and "Neutral" in our audio data. We include the average values for the metrics, as well as the differences between the different clip labels.

tions, we first combine a subset of them to form our estimated probability of negative affect. Thresholding is then applied to provide the final prediction. Our results indicate that the optimal configuration is the predicted probability for only "anger" and a threshold of 0.4. Using this information, we reach a weighted F1 score of 0.78 on our Negative and Neutral audio clips. See Appendix C for details.

The current speech emotion recognition models do not perform well on our task, as speculated. Anecdotally, when directly running classification on our audio clips using the model, many clips classified by us as "Neutral" are often classified as sad or disgusted.

Pause Detection: As established, prolonged pauses indicate the presence of foreign language anxiety and should be considered as a cue in our framework. We aim to develop automated metrics that identify user utterances with these pauses.

We devise three metrics for determining whether an audio clip fulfills the criteria for "Pauses":

1. **Silence Ratio:** The quotient of the total amount of silence in a clip and the clip length.
2. **Pause Rate:** The result of dividing the number of pauses by audio length.
3. **Average Pause Length:** The average length of pauses.

For computing these metrics, we equip our system with Silero-VAD (Silero, 2021), a fast and enterprise-grade voice activity detection package. Silero-VAD identifies and locates speech segments, and it allows speech extraction from the original audio such that the resulting clip is speech-only. We can therefore compute the total lengths of silence and pauses in an audio, as well as the number of pauses in an audio clip.

Other features, such as pause location, can also be used to indicate the level of anxiety. We leave the exploration of these features to future work.

To compare the ability of these metrics to differentiate between "Neutral" and "Pauses" clips, we calculate the values of these three metrics on these clips. We further measure the differences between the metric values for the two categories (Table 4). The "Average Pause Length" metric yields the highest difference, which suggests it effectively separates "Neutral" and "Pauses" clips. In addition, we experiment with various thresholds for differentiating the two types of audio using "Average Pause Length" (See Appendix B) and select a threshold of 0.5.

3.5 Empathetic Response Generation

Data Construction: Given the ASR transcripts of user utterances in a conversation, we added all instances of three consecutive utterances to our data (i.e., utterances 1+2+3, 2+3+4, etc). This created 2014 segments for optimizing our ChatGPT prompts. Due to cost constraints, we only used

625 conversation segments for prompt optimization: 125 for optimization, 200 for evaluation and iteration, and 300 for held-out testing.

Implementation: Our desiderata for the empathetic response generation module include the following: 1. Tailored to the user; 2. Empathetic and encouraging; 3. Including actionable feedback or specific examples the user can learn from. Because there are no sufficiently large datasets that precisely fulfill these requirements, we rely on prompting ChatGPT to generate such responses.

Unfortunately, large language models are sensitive to how they are prompted. Simple trial-and-error did not achieve consistently satisfactory responses in our preliminary experiments (ZEROSHOT stage).

We employed the DSPy framework (Khattab et al., 2023) to optimize for prompts while satisfying our desiderata (OPTIMIZED stage). We first tasked GPT-4 to check whether each requirement is satisfied in a given response (e.g. is the utterance empathetic and encouraging). This is a form of AI feedback (Bai et al., 2022). GPT-4 appears successful in this text annotation task, consistent with results established in Gilardi et al. (2023). Using the AI feedback as our *metrics*, we aimed to optimize our prompts to maximize the metrics. DSPy supplies the BayesianSignatureOptimizer, which references simple descriptions of our desiderata to suggest sample instructions and few-shot examples. Using this Bayesian-model-powered optimization process, we improved the metrics on a held-out test set from 68.3 (at the ZEROSHOT stage) to 89.8. We discuss whether the improvement aligns with human intuition in Section 5.1.

We observe that the outputs of our optimized prompt are often formal, while most of our design survey participants prefer colloquial feedback. To address this, we insert a final rewrite call to rewrite the optimized prompt output to a more colloquial version (REWRITE stage). GPT-4 evaluates this stage's outputs as 88.7.

During inference time, when we detect that the user requires empathetic feedback, the user's three most recent utterances are concatenated and fed into ChatGPT with the optimized prompt. The output undergoes the REWRITE stage to produce the final output. All ChatGPT prompts and GPT-4 feedback prompts used for this module are included in Appendix I. See Appendix J for examples of outputs at different stages.

3.6 Connecting Feedback and Conversation

User Query Response: Our feedback modules are currently intended for single-turn feedback (i.e. the bot provides the feedback without anticipating that the user will ask clarification questions), but in preliminary user studies, we noticed that users do inquire about the feedback. Therefore, we handle this case by constructing a ChatGPT call with the immediate conversation context and asking for a response to the user's query. We classify a user response to feedback as a relevant query with a rule-based approach. We use this rule-based approach instead of forwarding all post-feedback user queries to ChatGPT because prior users would ask about the bot's creator and training data, resulting in unintended behavior (e.g. the bot claiming it is created by Google or OpenAI engineers).

Transition: We employ templates for a smooth transition between feedback and the original conversation. Before entering the feedback stage, we cache the original bot response to return to the conversation afterward. More details about templates and ChatGPT prompts are provided in Appendix E.

3.7 Conversation

Unlike the other modules that only need to be activated sporadically, the conversation module is invoked for almost every turn. This poses additional needs for inference speed and costs, which motivates using a locally stored model.

We selected a Vicuna model fine-tuned for curriculum-driven conversations (Li et al., 2023). The model allows for customization of topics, chatbot personas, and vocabulary to incorporate into the conversation. Li et al. (2023) noticed that brevity instructions are sometimes ignored by ChatGPT. This further makes ChatGPT not ideal for our spoken conversation use-case, as run-on utterances may be difficult to comprehend in a speech setting. Users found the Vicuna model more helpful for developing conversational skills, providing natural and realistic utterances, and aligning with users' English proficiency levels.

The topic of "Name a movie that has had an enduring impact on you" was chosen for relatability. We randomly selected a vocabulary and one of the female personas to match the TTS voice. Bot feedback and user responses to feedback are not included in the conversation history when prompting the Vicuna model to keep the components modular and prevent out-of-distribution behavior.

	Quality	Conf.	Useful	Enc.	Listen	Care	Praise	PAS
Average	3.75	3.33	3.83	3.16	3.58	3.08	3.25	3.27
Std	1.05	1.07	1.19	1.64	1.16	1.24	1.60	1.16

Table 5: Post-survey results. "Conf." stands for confidence, "Enc." stands for encourage, and "PAS" stands for perceived affective support.

4 User Study

Fourteen native Mandarin speakers were recruited from social media and the authors' connections. Each participant conversed with the chatbot for at least 10 turns (a turn is one round of exchange between the chatbot and the user). A pre-survey for participant English proficiency and a post-survey for user experience were administered. In the pre-survey, we obtain an approximate assessment of the participants' English proficiency including their standardized test scores, self-reported proficiency, and the frequency at which they speak English daily. After the participants interacted with our system, they were presented with a post-survey which includes a modified version of the teacher affective support scale (the last four items below) (Sakiz, 2007) adapted for our context and general evaluations of conversation quality.

Our Likert-scale post-survey includes:
Quality: How was the conversation quality?
Confidence: Do you feel that you are more confident after conversing with the chatbot?
Useful: Do you think the chatbot's grammar feedback is useful?
Encourage: The chatbot encourages me when I am having difficulties in the conversation.
Listen: The chatbot listens to me when I have something to say.
Care: My opinion matters to the chatbot.
Praise: The chatbot recognizes and appreciates when I am good at something.

Details for the surveys can be found in Appendix H. Example conversations between the participants and the bot can be found in Appendix G.

5 Results and Discussion

5.1 Empathetic Generation Evaluation

We asked each participant to rank the different stages of empathetic feedback (ZEROSHOT, OPTIMIZED, REWRITE). Participants ranked responses generated in these three stages on the same segment for 30 randomly selected segments. At least 3 participants ranked each triple. We also asked the participants how they would improve the utterance they ranked at #1 for each conversation segment.

Stage	vs. ZEROSHOT	vs. OPTIMIZED	vs. REWRITE
ZEROSHOT	-	0.52	0.45
OPTIMIZED	0.47	-	0.45
REWRITE	0.54	0.54	-

Table 6: Win rates between each pair of empathetic feedback generation stages.

In Table 6, REWRITE wins more often against both ZEROSHOT and OPTIMIZED, suggesting that the REWRITE improves OPTIMIZED stage outputs. OPTIMIZED outputs are often not preferred due to their formality and length. Since REWRITE rephrases OPTIMIZED outputs without modifying core content, it appears that the participants are ranking the content from OPTIMIZED relatively higher than the content from ZEROSHOT. Another result is that ZEROSHOT is often ranked as #1 or #3, illustrating that ZEROSHOT outputs are less consistent in quality. Despite being scored higher by GPT-4, OPTIMIZED does not significantly outperform ZEROSHOT. This could be due to DSPy optimization focusing on fulfilling metrics without considering human preferences, or due to raters having various standards.

As for improving the feedback, participants reported that the best responses are still too verbose (one wrote "the shorter the better") and requested better feedback examples. They mentioned that generic praises can sound disingenuous, detrimental to the intention to encourage. Some suggested that praise may not be necessary for every piece of feedback, especially when participants receive multiple feedback during a conversation. One future direction would be to develop more context-aware mechanisms for more naturalistic and long-term empathetic feedback.

5.2 Conversation Statistics

Two participants did not receive empathetic feedback and were excluded from analyses. For the other twelve participants, each conversed for an average of 14.5 turns and received 1.9 grammatical feedbacks and 1.3 empathetic feedbacks.

5.3 Survey Results

On average, our participants have approximately 14.25 years of experience learning English. They all rated themselves above three out of five for self-reported English proficiency (higher is more proficient) with an average of 3.92. The participants who disclosed their IELTS and TOEFL scores had 7.3 and 109.3 averages respectively. For the question on English usage frequency, the average was 3.41 (one being for English only, five being for Mandarin only). Our participants have intermediate English proficiency but do not speak English frequently.

The post-survey results are shown in Table 5. In addition to the survey items, we include PAS as an aggregate metric by averaging the four adapted PAS survey items. The participants often consider the conversation quality to be high. They reported gaining moderate confidence after the conversation, and consider the bot's feedback useful. As for the survey items involving PAS, the results contain higher variance. While users believe that the bot appears to listen to them fairly attentively (potentially as an effect of the grammatical feedback), they are more ambivalent about whether the bot encourages them or praises them appropriately. We suspect that the reason for lower "Encouragement" ratings stems from our imperfect detection mechanism; empathetic feedback might have been given when the user was not exactly struggling. The participants also could not have struggled at all during the conversation. A potential reason for the high variance in "Praise" ratings is the disingenuous-sounding encouragement mentioned in Section 5.1. Additionally, user motivation for using our system can affect their self-reported results. Participants who only intend to test the system rather than improve their English might rate it poorly.

5.4 Dialogue Inspection and User Feedback

We inspect conversations with low PAS to identify failure modes of our system. The conversation with the lowest PAS includes both technical issues in the system (the user was baffled by the frequent interruptions in the system) and the chatbot forgetting the conversation history due to the limited context length of our model. Another conversation features significant ASR errors and the error propagation led to nonsensical grammatical feedback which confused the user. Due to current limitations in user query processing after bot feedback, some user queries were occasionally ignored, but the presence of these does not dictate low PAS.

We requested feedback from our participants. They praised the ASR accuracy and feedback quality, mentioning that they feel encouraged after receiving feedback. Some users stated that the goal of spoken English is to keep the conversation going, and therefore only egregious grammar errors should be corrected. Others would solicit grammar feedback from the system and exhibit dismay when it did not recognize their errors. One user mentioned that they would stammer and have disfluencies that would be recognized as grammatical errors. Some users disliked the stiffness of the feedback formats as they felt the conversation flows were interrupted. A subset of responses are presented in Appendix F.

These observations highlight limitations in our current system. To improve user experience, we will develop more seamless feedback mechanisms and robust user query classification. Additionally, we aim to create better grammar models suited for transcribed utterances and resilient to disfluencies and fillers. Additional goals include detecting technical difficulties so the chatbot can apologize for any interruption, as well as conversation summarizers to inform our model of previous discussions.

6 Conclusion and Future Work

In this work, we propose the negative emotion detection task in the context of English learning to capture learner frustration and anxiety. We also introduce the first English-teaching chatbot that provides adaptive, empathetic feedback to students using our negative affect detection pipeline. Initial trials with end users demonstrate the potential of our system. For future work, we intend to scale up our user evaluations and verify our hypothesis that our system can effectively improve student L2 grit.

For future work, apart from addressing participant feedback, we intend to expand our experiments to include more thorough comparisons between the different experimental conditions to establish more robust results. Specifically, we want to determine whether our adaptive empathetic feedback improves L2 grit more than no empathetic feedback or fixed feedback upon multiple chatbot interactions. Another interesting topic to examine more closely would be whether humans behave and react similarly when conversing with chatbots and real-life English teachers. We intend to include participants from an ESL course in our next study.

7 Limitations

Our current system serves as a proof-of-concept for a chatbot system capable of adaptive empathetic feedback, and it is by no means perfect. While our modular design allows for more rigorous control for future experiments, there can easily be error propagation between modules, and none of the modules are completely error-proof, as we have illustrated in our paper. To begin with, our speech emotion recognition pipeline does not successfully capture all instances of negative affect in our labeled data. The Llama model used for grammar correction still cannot correct all instances in the ErAConD test set. Our user query detection mechanism can miss relevant queries. All of these should be improved in future iterations of the system.

The current user study results are preliminary and do not offer sufficient statistical strength for solid conclusions. In future, we will aim for larger user studies by recruiting broadly on social media and at our institution.

Our data is currently labeled only by two labelers, which renders our labels less valid. We will aim to include more labelers to improve the validity of our emotion-labeled data.

8 Ethical Considerations

Any applications interfacing with humans, especially students, need to consider accidental psychological harm done to the students as a result of generations. To address this, we performed rigorous testing prior to our user study.

There is potentially self-identifying information present in our audio data. We will filter out self-identifying information before releasing the data to protect user identity.

References

Junyi Ao, Rui Wang, Long Zhou, Chengyi Wang, Shuo Ren, Yu Wu, Shujie Liu, Tom Ko, Qing Li, Yu Zhang, Zhihua Wei, Yao Qian, Jinyu Li, and Furu Wei. 2022. SpeechT5: Unified-modal encoder-decoder pre-training for spoken language processing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5723–5738, Dublin, Ireland. Association for Computational Linguistics.

Emmanuel Ayedoun, Yuki Hayashi, and Kazuhisa Seta. 2015. A conversational agent to encourage willingness to communicate in the context of english as a foreign language. *Procedia Computer Science*, 60:1433–1442.

Emmanuel Ayedoun, Yuki Hayashi, and Kazuhisa Seta. 2020. Toward personalized scaffolding and fading of motivational support in l2 learner–dialogue agent interactions: an exploratory study. *IEEE Transactions on Learning Technologies*, 13(3):604–616.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42:335–359.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Bridget Cooper. 2002. *Teachers as moral models: the role of empathy in the relationships between teachers and their pupils*. Ph.D. thesis, Leeds Metropolitan University.

Weihui Dai, Shuang Huang, Xuan Zhou, Xueer Yu, Mirjana Ivanovi, and Dongrong Xu. 2014. Emotional intelligence system for ubiquitous smart foreign language education based on neural mechanism. *Journal of Information Technology Applications & Management*, 21(3):65–77.

Sidney D'mello and Art Graesser. 2013. Autotutor and affective autotutor: Learning by talking with cognitively and emotionally intelligent computers that talk back. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2(4):1–39.

Luke Fryer, David Coniam, Rollo Carpenter, and Diana Lăpușneanu. 2020. Bots for language learning now: Current and future directions.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd-workers for text-annotation tasks. *arXiv preprint arXiv:2303.15056*.

Yoshiko Goda, Masanori Yamada, Hideya Matsukawa, Kojiro Hata, and Seisuke Yasunami. 2014. Conversation with a chatbot before an online efl group discussion and the effects on critical thinking. *The Journal of Information and Systems in Education*, 13(1):1–7.

Jonatas Grosman. 2021. Fine-tuned XLSR-53 large model for speech recognition in English. `https://huggingface.co/jonatasgrosman/wav2vec2-large-xlsr-53-english`.

Suleyman Kasap. 2019. Anxiety in the efl speaking classrooms. *The Journal of Language Learning and Teaching*, 9(2):23–36.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2023. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*.

Na-Young Kim. 2018. A study on chatbots for developing korean college students' english listening and reading skills. *Journal of Digital Convergence*, 16(8).

Na-Young Kim, Yoonjung Cha, and Hea-Suk Kim. 2019. Future english learning: Chatbots and artificial intelligence. *Multimedia-Assisted Language Learning*, 22(3).

John Kominek and Alan W Black. 2004. The cmu arctic speech databases. In *Fifth ISCA workshop on speech synthesis*.

Tony Chiu Ming Lam, Klodiana Kolomitro, and Flanny C Alamparambil. 2011. Empathy training: Methods, evaluation practices, and validity. *Journal of Multidisciplinary Evaluation*, 7(16):162–200.

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.

Yu Li, Shang Qu, Jili Shen, Shangchao Min, and Zhou Yu. 2023. Curriculum-driven edubot: A framework for developing language learning chatbots through synthesizing conversational data. *arXiv preprint arXiv:2309.16804*.

Kai-Hui Liang, Sam Davidson, Xun Yuan, Shehan Panditharatne, Chun-Yen Chen, Ryan Shea, Derek Pham, Yinghua Tan, Erik Voss, and Luke Fryer. 2023. ChatBack: Investigating methods of providing grammatical error feedback in a GUI-based language learning chatbot. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 83–99, Toronto, Canada. Association for Computational Linguistics.

Hao-Chiang Koong Lin, Ching-Ju Chao, and Tsu-Ching Huang. 2015. From a perspective on foreign language learning anxiety to develop an affective tutoring system. *Educational Technology Research and Development*, 63:727–747.

Liang-Chen Lin. 2014. Understanding pronunciation variations facing esl students. *International Journal of Humanities and Social Science*, 4(5):16–20.

Michael Pin-Chuan Lin and Daniel Chang. 2020. Enhancing post-secondary writers' writing skills with a chatbot. *Journal of Educational Technology & Society*, 23(1):78–92.

Steven R Livingstone and Frank A Russo. 2018. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one*, 13(5):e0196391.

Reza Lotfian and Carlos Busso. 2017. Building naturalistic emotionally balanced speech corpus by retrieving emotional speech from existing podcast recordings. *IEEE Transactions on Affective Computing*, 10(4):471–483.

Roy Lyster, Kazuya Saito, and Masatoshi Sato. 2013. Oral corrective feedback in second language classrooms. *Language teaching*, 46(1):1–40.

Michal Mazur, Rafal Rzepka, and Kenji Araki. 2011. Proposal for a conversational english tutoring system that encourages user engagement. In *Proceedings of the 19th International Conference on Computers in Education*, pages 10–12.

Susana Pérez Castillejo. 2019. The role of foreign language anxiety on l2 utterance fluency during a final exam. *Language Testing*, 36(3):327–345.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR.

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.

Gonul Sakiz. 2007. *Does teacher affective support matter? An investigation of the relationship among perceived teacher affective support, sense of belonging, academic emotions, academic self-efficacy beliefs, and academic effort in middle school mathematics classrooms*. Ph.D. thesis, The Ohio State University.

Olga C Santos, Mar Saneiro, Jesus G Boticario, and María Cristina Rodriguez-Sanchez. 2016. Toward interactive context-aware affective educational recommendations in computer-assisted language learning. *New Review of Hypermedia and Multimedia*, 22(1-2):27–57.

Nuobei Shi, Qin Zeng, and Raymond Lee. 2020. The design and implementation of language learning chatbot with xai using ontology and transfer learning. *arXiv preprint arXiv:2009.13984*.

Silero. 2021. Silero vad: pre-trained enterprise-grade voice activity detector (vad), number detector and language classifier. https://github.com/snakers4/silero-vad.

Yasser Teimouri, Luke Plonsky, and Farhad Tabandeh. 2022. L2 grit: Passion and perseverance for second-language learning. *Language Teaching Research*, 26(5):893–918.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jeffrey Dawala Wilang and Thanh Duy Vo. 2018. The complexity of speaking anxiety in a graduate efl classroom. *Journal of Asia TEFL*, 15(3):682.

Chih Hung Wu, Hao-Chiang Koong Lin, Tao-Hua Wang, Tzu-Hsuan Huang, and Yueh-Min Huang. 2022. Affective mobile language tutoring system for supporting language learning. *Frontiers in Psychology*, 13:833327.

Wangjiao Wu, Yabing Wang, and Ruifang Huang. 2023. Teachers matter: exploring the impact of perceived teacher affective support and teacher enjoyment on l2 learner grit and burnout. *System*, 117:103096.

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, Seattle, United States. Association for Computational Linguistics.

Chunpeng Zhai and Santoso Wibowo. 2022. A systematic review on cross-culture, humor and empathy dimensions in conversational chatbots: The case of second language acquisition. *Heliyon*.

Dian Zhang. 2022a. Affective cognition of students' autonomous learning in college english teaching based on deep learning. *Frontiers in psychology*, 12:808434.

Zhichao Zhang. 2022b. Toward the role of teacher empathy in students' engagement in english language classes. *Frontiers in Psychology*, 13:880935.

Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. 2020. The design and implementation of xiaoice, an empathetic social chatbot. *Computational Linguistics*, 46(1):53–93.

A Chatbot Design Discussion

A total of 12 Mandarin native speakers were recruited through the authors' personal connections to provide feedback on our chatbot design. We provided a Google Form for them to complete. We later released a version on social media that gained many more responses, but since we do not utilize the results from that survey directly in this work, we reserve the discussion and incorporation of these results for future work.

We translated a subset of relevant questions and response options from Mandarin. We have abridged preambles in the questionnaire for brevity. The questions and responses of our survey are as follows (the percentage in the parentheses corresponds to the percentage of participants who selected that option):

1. *How would you like an English teacher to give you feedback?*
   - Formal (25%)
   - **Colloquial (75%)**
2. *How long should the feedback be?*
   - 1 - 2 sentences (8.3%)
   - 2 - 3 sentences (41.7%)
   - **3 - 4 sentences (50%)**
   - 4+ sentences (0%)
3. If you have made a grammatical error, what specific attributes should a teacher's feedback for you have? Please select all that are applicable.
   - Correct your mistakes directly (58.3%)
   - Correct your mistakes interactively through Q & A (33.3%)
   - **Give you examples so that you can learn from the example and avoid making the same mistakes again (75%)**
4. What is your ideal form of encouraging and empathetic feedback? Please select all that apply.
   - Give you encouragement (e.g. "You are doing great!", "I am proud of you!") (58.3%)
   - **Tell you what you are good at in terms of your spoken English (75%)**
   - Tell you what you can improve in your spoken English (58.3%)

Threshold	Neutral%	Pauses%
0.1	100.0	3.5
0.2	98.1	22.5
0.3	72.2	53.0
0.4	44.4	72.5
0.5	26.0	85.5
0.6	7.4	92.0
0.7	3.7	97.0

Table 7: Classification accuracy for Neutral and Pauses audio clips using the Silence Ratio pause detection metric.

Threshold	Neutral%	Pauses%
0.1	100.0	0.0
0.2	98.1	1.5
0.3	96.3	9.5
0.4	88.9	18.5
0.5	74.1	39.5
0.6	50.0	61.0
0.7	29.6	85.0

Table 8: Classification accuracy for Neutral and Pauses audio clips using the Pause Rate pause detection metric.

- **Tell you how you can improve through examples (75%)**
- Provide you with plausible and actionable advice for improving your English (41.7%)

B Experiments for Pause Detection

After computing the pause length metric values for all audio clips labeled as either "neutral" or "pauses", we compared the effect of varying pause detection mechanisms and thresholds. We computed the classification accuracy values when using different pause detection metrics and different threshold values. We aim to obtain the highest possible classification accuracy values for our metric-threshold combination.

We present the results for varying threshold values for Silence Ratio, Pause Rate, and Average Pause Length in Tables 7, 8, and 9, respectively.

C Experiments for Negative Emotion Detection

The wav2vec model we have selected outputs probabilities for the following emotions given a speech segment: angry, calm, disgust, fearful, happy, neutral, sad, and surprised. This model is a fine-tuned version of Grosman (2021), which is a widely popular automatic speech recognition model. The model is then fine-tuned on the RAVDESS dataset (Livingstone and Russo, 2018) for the speech emotion recognition downstream task.

Threshold	Neutral%	Pauses%
0.1	98.1	21.5
0.2	98.1	24
0.3	96.3	34.5
0.4	88.9	54.4
0.5	**70.4**	**73.0**
0.6	61.1	81.5
0.7	38.9	89.5

Table 9: Classification accuracy for Neutral and Pauses audio clips using the Average Pause Length pause detection metric.

We explored the following methods for combining the output probabilities from the model to produce the negative affect estimate. Here, we include anger for each of our methods because frustration and anger can manifest themselves in a similar manner in speech.

1. Anger + Disgust + Fearful + Sad (ADFS) - 0
2. Anger + Disgust + Fearful (ADF) - 1
3. Anger + Disgust (AD) - 4
4. Anger + Fearful (AF) - 5
5. Disgust + Fearful (DF) - 3
6. Anger (A) - 2

For the values of the threshold, we experimented with 0.1 through 0.9 with an increment of 0.1.

We evaluated the different setups on all audio clips labeled as Neutral or Negative in our labeled data subset. The weighted F1 score was computed to account for class imbalance. We included the best F1 scores achievable by each setup, along with their corresponding thresholds for obtaining the best F1 scores, in Table 10.

D Training Details for Llama and Flan-T5 Models

All models were trained on a single 40 GB GPU. Models were trained for 10 epochs and the best models were selected using validation set loss.

Setup	Threshold	Best F1
ADFS	0.9	0.57
ADF	0.8	0.76
AD	0.8	0.76
AF	0.4	0.76
DF	0.8	0.76
A	0.4	**0.78**

Table 10: The best achievable weighted F1 scores and their corresponding threshold values for each of the speech recognition model output aggregation methods.

Flan-T5-base was trained without any parameter-efficient fine-tuning, but all other models were trained using PEFT and Lora. We will release model training and inference code if accepted.

E Details on the Connecting Feedback and Conversation Component

E.1 Identifying Query

We utilized a simple rule-based approach to determine whether an utterance provided by a user after bot feedback is a question about the feedback or the English learning process. Namely, we (1) checked that a "?" is in the transcribed utterance; (2) checked whether one of the words in this list belongs in the utterance: "grammar", "grammatical", "vocab", "English", "mistake", "example", "sentence". If both conditions were fulfilled, we then interfaced with ChatGPT to respond to user queries.

E.2 Responding to Query

Since we anticipate that the user will only be responding to the bot feedback, we would not need to include too much context in our ChatGPT call. We supplied the following prompt to ChatGPT to obtain a response to give to the user.

```
f"""Based on the following conversation
    history:\n\n{convo}, answer the user's
    following query: "{user_query}" Answer
    in a spoken utterance. Provide specific
    feedback, but be succinct."""
```

E.3 Transitioning

If the user does not respond to the bot feedback with a query, or when the bot has finished responding to the user query, the system would then return to the original conversation flow. This transition was obtained by prefixing the cached original bot response with one of the randomly selected phrases. There are two general scenarios here:

1. The user expresses gratitude by including "thank" in their utterance.
2. The user does not explicitly express gratitude.

The code for constructing the prefix to prepend to the original bot response is as follows:

```
if "thank" in text.lower():
    prefix = random.choice(["Of course!",
         "No problem at all.", "Yeah, no
        problem!", "No problem!"]) + " "
        + random.choice(["Back to the
        conversation.", "Back to our
        convo.", "Let's go back to
        chatting.", "Now we circle back
        ."])
else:
    prefix = random.choice(
        ["Sounds great.", "Alright, let's
            continue our conversation.",
            "Great, let's get back to it
            !",
         "Okay let's go back to our
             conversation.", "Now back to
              our conversation.", "Okay!",

         "Lets' go back to our chat.", "
             Let's keep chatting."])
```

F User Feedback

We asked participants to share their experience with the chatbot, and received the following responses. All responses have been paraphrased and translated from Mandarin:

User A: *I like the chatbot and I look forward to the finished product... When the chatbot corrects my grammar, the way it corrects me is very stiff and clunky, and it can interfere with the immersive experience... Additionally, when I use filler words or discourse markers or perform self-repair in the conversation, the bot would catch these as grammatical errors even though they should not be treated as such.*

User B: *Your chatbot has high ASR accuracy, and the quality of the conversation is quite good! However, it may be challenging to provide English learners with sufficient help, because I asked for more specific feedback after receiving a feedback and it went back to talking about movies.*

User C: *One thing I really like about the chatbot is that the feedback is very timely! When using ChatGPT I don't always get very prompt feedback. We have a foreigner English teacher coming next week and your bot is definitely a life-saver.*

User D: *I don't think some trivial grammatical errors, like mixing is and was when discussing movie plots, should be corrected. I also asked for the bot for word choice improvements but it didn't give me any.*

User E: *The conversation has high quality, and although the feedback can lag behind one or two sentences, the feedback is always correct and helpful. The bot clearly understands what you are trying to convey and can continue the conversation despite grammar errors. Sometimes it uses sentences that can be too complex for beginners, but providing Mandarin translations for bot utterances can be helpful.*

G Example Conversations

The following are conversation excerpts between our bot and different users. Here, blue indicates empathetic feedback, orange indicates a transition between feedback stage and the conversation, and red indicates a grammatical feedback.

We first examine a successful conversation example. Note that the empathetic feedback is activated when the user experiences hesitation:

User: Actually, I hardly ever watch movies, so could I describe opera?

Bot: Sure! What's the name of the opera that you'd like to describe?

User: Okay, that's Turandot, which describes a love story between a Chinese princess and a foreign prince.

Bot: I believe you wanted to say "Okay, that's Turandot, which is a story about a love between a Chinese princess and a foreign prince.". In this sentence you used the verb "describes", but it doesn't sound natural. I'd recommend using the word "is a story about". Does that sound good?

User: you

Bot: Okay let's go back to our conversation. Interesting! I can't say that I'm familiar with it. Could you tell me more about the story?

User: This Chinese princess whose grandma is... Wait a minute.

Bot: You've got a good grasp of the topic and can explain yourself clearly, which is awesome! Just tweak your grammar and sentence structure for a more natural flow. For instance, instead of saying "I hardly ever watch movies, so could I describe opera?", try "Since I rarely watch movies, can I talk about opera instead?" Keep at it, and you'll see improvement in no time! Does that sound alright to you?

User: Yes, and now I will go to talk about the context of the opera.

Bot: Okay! Yes, go on. What's the story about?

User: This story is about the Chinese princess, Truong Du, whose grandma was bullied by the foreigners. So Truong Du set a rule to the man who want to marry him that he must answer three questions and then he can marry her or dad. The cover of the, answered the three questions and finally married with Torandu.

Bot: Perhaps what you meant to say was "who wants " and not want In this sentence you made a mistake on the verb "want". The correct verb form here is "wants". Remember to make your verbs agree with their subjects. How does that sound?

User: That sounds great, okay I understand.

Bot: Alright, let's continue our conversation. That's certainly a unique story! What did you like most about the opera?

Here we see a failure of recognizing user query due to the rule-based nature of our query recognition mechanism.

Bot: I'm doing alright, thanks. Have you watched any good movies lately?

User: No, not really. I don't watch any movie.

Bot: Really? You don't watch any movies? Not even the classics?

User: then it won't be lately. I do watch classics. Which one you want to talk about?

Bot: Well, I was thinking about the movie "Citizen Kane". What do you think about it?

User: That is a very very classical one. I had it some time ago I couldn't really remember the plot could you rephrase it for me?

Bot: Sure, it's about a journalist who becomes a powerful newspaper publisher and his journey to find the meaning of "rosebud" which holds the key to his soul.

User: Yeah, I sort of like that idea. You know, start being a nobody and then with one's effort you became a big shot in the industry.

Bot: Exactly, it's a great story of how one's perseverance and hard work can lead to success. I also love the way the director used different narrative techniques to tell the story.

User: I couldn't remember exactly. Could you give me an example?

Bot: Sure, one of the techniques used is the use of different reporters telling the story of Kane's life. Each one provides a different perspective on who

he was as a person and the events that shaped his life.
User: Yeah I would say that was quite innovative at that time when the movie was first released. Anyway what else you want to talk about?
Bot: You're doing great with understanding and expressing the topic clearly! Just brush up on grammar and sentence structure, like saying "I can't remember exactly." Keep practicing to keep getting better!
User: All right, anything else for me?
Bot: Lets' go back to our chat. Well, I was also thinking about "The Godfather". Have you seen it?
User: Oh yes, I really love the music. The Godfather theme in both 1 and 2. I didn't really watch Godfather, the third part.
Bot: I believe you wanted to say "I didn't really watch The Godfather, the third part.". You seem to be missing a determiner in this sentence. You should probably add "The" to make the sentence sound more natural. Does that sound alright to you?

H Pre Survey and Post Survey

The surveys have been translated from Mandarin.

H.1 Pre-Survey

1. How many years have you been learning English?

2. How would you rate your English proficiency (1 star for beginner, 5 stars for proficient)?

3. What is your IELTS score?

4. What is your TOEFL score?

5. How much English do you speak every day (1 star for English only, 5 stars for Mandarin only)?

H.2 Post-Survey

1. How was the quality of your conversation on a scale of 1 to 5?

2. Do you feel that you are more confident after conversing with the chatbot? Rate on a scale of 1 to 5.

3. Do you think the chatbot's grammar feedback is useful? Rate on a scale of 1 to 5.

4. The chatbot encourages me when I am having difficulties in the conversation (1-star for "Not like me at all", 5-stars for "Very much like me").

5. The chatbot listens to me when I have something to say (1-star for "Not like me at all", 5-stars for "Very much like me").

6. My opinion matters to the chatbot (1-star for "Not like me at all", 5-stars for "Very much like me").

7. The chatbot recognizes and appreciates when I am good at something (1-star for "Not like me at all", 5-stars for "Very much like me").

I Empathetic Feedback Generation ChatGPT Prompts

I.1 Zero-shot Prompt

```
A student is learning English. You are assessing
     a spoken utterance. In at most two
    sentences, summarize (1) their specific
    strengths in English skills and (2) things
    they can work on to improve. Address the
    student in the second person. Include
    specific examples that the student can learn
     from. Be colloquial, as if in spoken
    conversation.

---

Follow the following format.

Convo: \$\{convo\}
Reasoning: Let's think step by step in order to
    \${produce the output}. We ...
Output: Treat this as a spoken conversation, so
    be succinct, colloquial, and empathetic.

---

Convo: - Yeah, I always talk to my parents about
     the student. - Yeah, so the person that I
    met, you know, that I've met, that I want to
     study with is one of my classmates, Tom,
    and how I met this, met Tom is when I was in
     high school and he's in my next classroom
    and he is one of the best students in the
    other class and I really admire his
    intelligence and I learn a lot from him and
    I've known him since, you know, 2016 and he
    is one of the best students in the other
    class so that's why I want to study with him
    .
Reasoning: Let's think step by step in order to
    assess your spoken utterance. We can see
    that you have a strong grasp of English
    vocabulary and grammar, as evidenced by your
     ability to express complex ideas and use a
     variety of sentence structures. However, you
      may want to work on your pronunciation and
     intonation, as some of your words were not
     clear and your speech lacked natural rhythm.
      For example, you said "met" instead of "
     meet" and "classroom" instead of "classmate
     ." Practicing with a native speaker or using
      online resources can help you improve in
     this area. Keep up the good work!
Output:
```

I.2 Optimized Prompt

```
Proposed Instruction: You're playing the role of
      an encouraging English tutor for a student
     who is actively learning and practicing
     their English through conversation. Your
     task is to listen attentively to their
     spoken utterances and provide constructive
     feedback. In your response, kindly highlight
      (1) one specific strength they showed or an
      aspect they did well in during the
     conversation, complimenting their effort or
     skill in English, and (2) offer one focused
     suggestion on how they can improve further,
     making it actionable and clear. Use
     colloquial language to maintain the
     conversational tone, directly addressing the
      student with "you", and where possible,
     reference specific examples from their
     speech to illustrate your points. Your
     feedback should feel like a supportive nudge
      towards their language learning journey,
     keeping it concise and personalized.

---

Follow the following format.

Convo: ${convo}
Reasoning: Let's think step by step in order to
     ${produce the output}. We ...
Feedback: Treat this as a spoken conversation,
     so be succinct, colloquial, and empathetic.

---

Convo: - Sorry, I have not get some information
     about the personal medical data sharing.
     Well, I think it's maybe something related
     to digital. Like, it's different from the
     traditional medical system. Like, all
     medical systems transfer from the
     traditional one into the digital one. And
     maybe the digital systems will help the
     hospital to organize the patients and to
     regulate some patient's information with
     high efficiency and better orders. So it may
      saves a lot of time for some... - Yeah,
     maybe like if you go to one hospital and if
     the doctor keeps a capital record on his on
     the paper then you left this hospital go to
     another hospital the new doctor have no idea
      about your situation if all information of
     patients can put on the website or in other
     words those are online information that
     whether we change the hospital the doctor
     can get access to our information and we can
      show our situation to to those doctors yeah
      it's a good idea - Yeah, if we put some
     information online, it means we lose our
     Privacy or our information is easy to be
     steered by some Bad person. Yeah, maybe some
      people they have some Some disease which
     they don't want to be known They want to
     keep at a secret if we upload their disease
     Their information about this disease some
     people there will feel uncomfortable maybe
     we can like set some System which only
     allowed the The doctor can see all this
     information Information something like the
     bank so people cannot get a sex to the
     system if they are not Permated permitted to
      do these things
Reasoning: Let's think step by step in order to
     assess your spoken utterance. You have a
     strong understanding of the topic and are
     able to explain it clearly. However, you can
      work on your pronunciation and grammar to
     improve your fluency. For example, you can
     practice pronouncing words like "information
     " and "permitted" more clearly and work on
     using proper verb tenses. Keep practicing
     and you will continue to improve!
Feedback: You have a good grasp of the topic and
      can explain it well. However, you can work
     on your pronunciation and grammar to improve
      your fluency. For example, try practicing
     words like "information" and "permitted" and
      using proper verb tenses. Keep practicing
     and you'll get even better!

---

Convo: - The people who influence me the most
     was my primary teacher, Ms Wong. She made me
      follow the English teaching class on the
     road to the next level. She is so gentle, so
      calm, so professional. And the method she
     is teaching, she is interesting. So in her
     English class, my attention is always
     focused. She makes me feel that English is a
      very interesting subject. And she always
     shares with me the experience in English
     that she is in the world, her life, her
     study life and the normal life in the world.
      And that made me think that English is a
     necessary and important skill. And it played
      an important role in our life. - Yes, I
     told her, I always, when I chat with my
     friends, I will always mention this teacher.
      She, although she is the primary English
     teacher for me, is nearly 17 years ago. Yeah
     , it's a long time and we haven't had a long
      time to say that. I still remember her and
     always remember what she did for us. - I
     think the childhood is the most... It's the
     most pure and carefree days. In that time
     happiness is the very easy. And we always do
      what we like. We don't need to concern our
     life, concern how to make money for living.
Reasoning: Let's think step by step in order to
     assess your spoken English skills. You have
     a strong foundation in English, as evidenced
      by your ability to express yourself clearly
      and fluently. However, you can work on your
      pronunciation and intonation to make your
     speech more natural and engaging. For
     example, you can practice speaking with a
```

native speaker or listen to English podcasts to improve your pronunciation and intonation. Keep up the good work and continue to practice, and you will see improvement in no time!

Feedback: You have a great foundation in English, and it's clear that you have a strong grasp of the language. However, you can work on your pronunciation and intonation to make your speech sound more natural and engaging. For example, you can try practicing with a native speaker or listening to English podcasts to improve your pronunciation and intonation. Keep up the good work and keep practicing, and you'll see improvement in no time!

Convo: - Yes, of course. I usually focus on my test work. - When I was reading books or I'm doing some hard work. - I will find some help, maybe from another people or from folks.

Reasoning: Let's think step by step in order to assess this spoken utterance. We can see that you have a good grasp of basic English skills, such as using the correct verb tense and sentence structure. However, you may want to work on your vocabulary and pronunciation, as some words were mispronounced and you used "folks" instead of "people" or "others."

Feedback: You have a good understanding of basic English skills, like using the right verb tense and sentence structure. However, you may want to focus on expanding your vocabulary and improving your pronunciation. For example, instead of saying "folks," you can use "people" or "others." Keep practicing and seeking help from others to improve your skills.

Convo: - Well, as a way of recording things, I think it's much simpler and easier for you to take photos than writing a journal or something like that. - No, not really, because back then a few decades ago I think going to cinema was really expensive and we cannot afford it when I was a kid. - I prefer go to cinema with my friends because I don't have quite the same taste for movie or films with my parents so our friends is much better.

Reasoning: Let's think step by step in order to assess your spoken English skills. We can see that you have a good grasp of vocabulary and can express your thoughts clearly. However, you can work on your grammar and sentence structure to make your speech more fluent and natural. For example, instead of saying "I prefer go to cinema", you can say "I prefer going to the cinema". This will make your sentence more grammatically correct and sound more natural. Keep practicing and you will improve in no time!

Feedback: You have a good grasp of vocabulary and can express your thoughts clearly, which is great! However, you can work on your grammar and sentence structure to make your speech more fluent and natural. For example, instead of saying "I prefer go to cinema", you can say "I prefer going to the cinema". Keep practicing and you will improve in no time!

Convo: - Small businesses like small pop and man shops that people would open, small restaurants. I think those are pretty, you know, the go-to small business that one can think of on top of their minds. - I think I would prefer big companies because big companies are more insured, the quality is more controlled and if you have any issues with the product you have easier time to get a refund or replacement for small companies . I do understand that they have higher operation costs so sometimes they would try not to have refunds and replacements and so on. So it's understandable but as a consumer I would prefer big companies with their insurance policies so I think that is my choice. - Well, I think it's better to live it to work in a big company because in big companies you have a bigger platform. You have chances to learn from people who came before you, who were also working in the company. You have chances to work across companies because you have a lot of business dealing with them. And big companies often give their staff training with something that most small companies lack. So at least at the beginning stage of your career, I think working in a big company is definitely a better way. They're the better choice for most people.

Reasoning: Let's think step by step in order to assess your spoken utterance. You have a strong understanding of the differences between small and big businesses, and you provide specific examples to support your points. However, you can work on your pronunciation and intonation to make your speech more clear and engaging. For example, try to enunciate each word clearly and vary your tone to emphasize important points. This will help you communicate your ideas more effectively.

Feedback: You have a good grasp of the differences between small and big businesses and you give good examples to support your points. But you can work on your pronunciation and intonation to make your speech more clear and engaging. For example, try to say each word clearly and change your tone to emphasize important points. This will help you communicate your ideas more effectively.

Convo: - I like to listen to popular music. It is easy to follow. It's easy to sing. I like to share those favorite songs with my friends together. And when I listen to some music or some songs that sound so good, I

```
     like to listen to it again and again. - My
     favorite singer is Joe Shen. He is a male
     singer. His voice is similar to a female
     singer's voice. His songs are clean and
     clear and he can sing all kinds of songs. He
      can speak eight languages. It's amazing. I
     really like his songs. - Yeah, I do like art
     . I like to, I like those pictures,
     paintings. I really want to draw some
     pictures by myself but I'm not good at it. I
     'd like to see them.
Reasoning: Let's think step by step in order to
```

I.3 Rewrite Prompts

There are two stages to the rewrite process. In the first stage, we employed the following prompt:

```
Shorten and rewrite this utterance to sound
     simple, natural, and engaging; remove any
     assessment of speech including pronunciation
      and intonation:\n\n{empathetic_output}
```

Here, `empathetic_output` refers to the output from the OPTIMIZED stage. We then continued appending the following message to the current session with ChatGPT:

```
Make your response different and casual, and
     shorten to 3 - 4 sentences
```

The output from ChatGPT after this additional prompting was used as the final output.

J Example Outputs at Different Stages

The following conversation segments are selected at random and may contain ASR errors.

Conversation Segment:

1. To be honest, I don't want to live on another planet because I think the earth is comfortable and it is suitable for us.
2. A little. I have some knowledge about the moon and the moon.
3. When I was a student in primary school, I went to... I often go to a picnic with my friends.

Zeroshot: You have a good grasp of the topic and can express your thoughts clearly. However, you can work on your fluency and pronunciation to make your speech more natural and easy to understand. For example, you can practice speaking more slowly and enunciating your words clearly. Also, try to listen to native speakers and imitate their pronunciation to improve your own. Keep up the good work!

Optimized: You have a good understanding of the topic and can express your thoughts clearly, which is great! However, you can work on your grammar and sentence structure to make your speech more fluent and natural. For example, instead of saying "I went to... I often go to a picnic", you can say "I used to go on picnics with my friends when I was a student in primary school." This will make your sentence more grammatically correct and sound more natural. Keep practicing and you will improve in no time!

Rewrite: You've got a good handle on the topic and express yourself clearly, which is awesome! Just tweak your grammar and sentence structure for a smoother and more natural flow. For example, instead of saying "I went to... I often go to a picnic," try "I used to go on picnics with my friends back in primary school." Keep at it, and you'll get even better in no time!

Conversation Segment:

1. Well, probably like less diet, less fat tissues would really be beneficial to your health and to your general functions.
2. What's the word again? I've never heard of it. Could you explain the terminology to me?
3. Okay, that sounds... Well, I'm asking you the word you're talking about the triple action capillates. Yeah, but what's that?

Zeroshot: You have a good understanding of vocabulary and sentence structure, but you may need to work on your pronunciation and listening skills. For example, when you didn't understand the word "capillates," you asked for clarification, which is great. To improve, try practicing speaking and listening to English more often. Keep asking for clarification when needed, and you'll continue to improve.

Optimized: You have a good understanding of the topic and can express your thoughts clearly, which is great! However, you can work on your vocabulary and pronunciation to improve your fluency. For example, instead of saying "less diet," you can say "a healthier diet" and instead of saying "capillates," you can say "capillaries." Keep practicing and expanding your vocabulary to improve your skills.

Rewrite: You've got a good grasp of the topic and can explain things clearly, which is awesome!

Just work on your vocab and pronunciation for smoother conversation. Try swapping “less diet” for “healthier diet” and “capillates” for “capillaries.” Keep at it and keep expanding your vocab to get even better!

Conversation Segment:

1. Hmm, actually it depends. If I’m concentrating on the things that I’m interested in, I won’t feel bored. But if the things just go so embarrassed or so boring, I will distract by something else. And another situation is that I may fall asleep.

2. Well, I actually do not have a memory of that. Because I think sitting on the floors may be the things that the boys do. And however, you know, I’m a girl who likes a tiny and quiet place to sit.

3. Of course, while speaking of a person with these strong impression, I would say that was my Chinese teacher in my primary school. I remember she taught us in my fifth grade. At my first sight, she looked really irradible, which made me have the feeling of this teacher is the person you can’t get along well with. So I always keep the distance with her. However, what upset me the most was right there. These schools always don’t allow the students to bring any entertainment things. I was just a kid, so I bought some comics to my friend as well as some classmates and spread it. Well, the comics are really interesting. After my Chinese teacher, who is also my class class teacher, knew that she just threw it out of my classroom window in front of all my classmates. And that was really embarrassing. So that memory kept kept a long time in my mind and I don’t really like that teacher. Even now, I can still remember her with not really good memories.

Zeroshot: You’re doing great in English, you can express yourself well and your thoughts are clear. But you can work on your pronunciation and intonation to make your speech more engaging and natural. Try practicing in front of a mirror or recording yourself to identify areas for improvement. Keep it up!
Optimized: You have a good understanding of the topic and can express your thoughts clearly, which is great! However, you can work on your grammar and sentence structure to make your speech more fluent and natural. For example, instead of saying "I remember she taught us in my fifth grade", you can say "I remember she taught us when I was in fifth grade." Also, try to use more descriptive words to make your story more engaging. Keep practicing and you will continue to improve!
Rewrite: You’ve got a good grasp of the topic and express yourself clearly, awesome! Just tweak your grammar and sentence structure for a smoother flow. For example, instead of saying "I remember she taught us in my fifth grade," you could say "I remember she taught us when I was in fifth grade." Keep practicing, and you’ll keep improving!

Conversation Segment:

1. People are often willing to get to know new people in various situations. Firstly, when individuals join new social or professional environments, such as starting a new job or moving to a new city, they are more open to meeting new people as they seek to establish social connections and build a support network. Secondly,

2. People go to various places to meet new people and expand their social circles. One common place is social events and gatherings such as parties, social clubs and community events. These provide opportunities for individuals to interact with others who share similar interests or are part of the same community.

3. Studying a conversation can be approached in various ways depending on the context and the individuals involved. One common way is through initiating a topic of mutual interest. This could involve asking open-ended questions or making observations about the current situation. For example, at a social gathering, one might start a conversation by commenting on the events, complimenting someone’s attire, or discussing a shared hobby.

Zeroshot: You’re great at starting conversations and finding common interests with others, which is a valuable skill in making new connections. However, you can work on being more confident and assertive in your approach, as well as actively listening and asking follow-up questions to keep the conversation going. For example, instead of just commenting on the events, try asking others about their experiences or opinions. This will show your interest and help you build deeper connections with people.

Optimized: You have a good understanding of the topic and can explain it clearly. However, you can work on your pronunciation and intonation to make your speech more engaging and natural. For example, try to vary your tone and emphasize important words to make your speech more interesting. Keep practicing and you will continue to improve!

Rewrite: You've got a good grip on the topic and explain it well. To spice up your speech, try varying your tone and emphasizing important words. Keep practicing, and you'll keep getting better!

Conversation Segment:

1. Well, I'm afraid that I can't give you the straight answer. But in the city where I live now, the weather is always raining because it's located in the southeastern part of China and it's also close to the sea. Sometimes I like to stay in the sunny days because I really like the moment when I read some books in the afternoon in the sunny days.

2. Well, there are some different kinds of styles in terms of weather situations in my hometown. For instance, in spring, the air is mild and fresh. In summer, the weather is so hot. In autumn, the weather started to become a little cold compared with the weather in summer. In winter, the weather in my hometown is quite cold.

3. Well, I'm afraid that I can't give you the straight answer. This mainly depends on different situations. When the weather is so dry, so when it's raining, I really like the rainy days. On the contrary, as I mentioned before, I really like the time when I spent in the sunny days like reading a book when I felt very tired.

Zeroshot: Hey there! Your pronunciation and intonation are on point, making it easy for me to understand you. But to make your speech more interesting, try using different words and sentence structures. For example, instead of saying "I really like", you can say "I'm fond of" or "I'm a big fan of". Also, use more descriptive adjectives like "scorching" or "sweltering" to paint a clearer picture. Keep practicing and you'll see improvement in no time!

Optimized: You have a good understanding of different weather situations and can describe them clearly, which is great! However, you can work on your grammar and sentence structure to make your speech more fluent and natural. For example, instead of saying "I really like the time when I spent in the sunny days", you can say "I really enjoy spending time in the sunny days." Keep practicing and you will continue to improve!

Rewrite: You're good at understanding different weather situations and describing them clearly, that's awesome! Just work on grammar and sentence structure to make your speech flow better. For example, say "I love spending time on sunny days" instead of "I really like the time when I spent in the sunny days." Keep practicing to improve even more!

Beyond Flesch-Kincaid: Prompt-based Metrics Improve Difficulty Classification of Educational Texts

Donya Rooein[1], **Paul Röttger**[1], **Anastassia Shaitarova**[2], **Dirk Hovy**[1]

[1]Bocconi University, [2]University of Zurich
{donya.rooein, paul.rottger, dirk.hovy}@unibocconi.it,
anastassia.shaitarova@uzh.ch

Abstract

Using large language models (LLMs) for educational applications like dialogue-based teaching is a hot topic. Effective teaching, however, requires teachers to adapt the difficulty of content and explanations to the education level of their students. Even the best LLMs today struggle to do this well. If we want to improve LLMs on this adaptation task, we need to be able to measure adaptation success reliably. However, current STATIC metrics for text difficulty, like the Flesch-Kincaid Reading Ease score, are known to be crude and brittle. We, therefore, introduce and evaluate a new set of PROMPT-BASED metrics for text difficulty. Based on a user study, we create PROMPT-BASED metrics as inputs for LLMs. They leverage LLM's general language understanding capabilities to capture more abstract and complex features than STATIC metrics. Regression experiments show that adding our PROMPT-BASED metrics significantly improves text difficulty classification over STATIC metrics alone. Our results demonstrate the promise of using LLMs to evaluate text adaptation to different education levels.

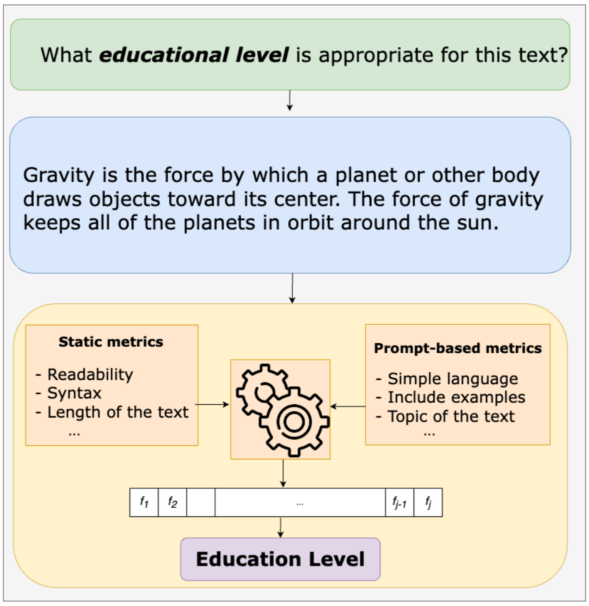

What ***educational level*** is appropriate for this text?
Gravity is the force by which a planet or other body draws objects toward its center. The force of gravity keeps all of the planets in orbit around the sun.
Static metrics
- Readability
- Syntax
- Length of the text
...
Prompt-based metrics
- Simple language
- Include examples
- Topic of the text
...
f_1
f_2
...
f_{j-1}
f_j
Education Level

Figure 1: Schematic overview of our approach to text difficulty classification. We calculate relevant STATIC and PROMPT-BASED metrics for a given input text. Either or both metrics are then fed into a regression classifier that makes a final classification.

1 Introduction

Large language models (LLMs) today can answer wide-ranging questions and explain complex concepts with high accuracy (Chung et al., 2022; OpenAI, 2023). This development has motivated explorations into their uses for education, ranging from automated student assessment and personalised content to dialogue-based teaching (Upadhyay et al., 2023; Sallam, 2023; Yan et al., 2023; Hosseini et al., 2023).

Effective teaching requires that the difficulty of content and explanations is tailored to the education level of the students. Human teachers are trained to do this, and adjust their material and style without much prompting. However, this adaptation is not just the adjustment of one variable. It is a complex undertaking, touching upon lexicon, syntax, pragmatics, and semantics. Improving the ability of LLMs to adapt their outputs to different levels of education is therefore crucial to unlocking their usefulness for education. One of the most basic requirements to achieve this goal is a way to measure adaptation success.

Measuring whether a given output is appropriate for a given level of education, however, is a very difficult task. Existing STATIC metrics, like the Flesch-Kincaid Reading Ease score (Flesch, 1948), are based on simple formulas, heuristics, and word counts. They share the brittleness of all heuristic approaches and are known to be noisy measures of text difficulty at best. Also, these metrics were developed for longer-form explanations, like those found in textbooks, rather than dialogue-style teaching. Due to their reliance on counts, their estimates

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 54–67
June 20, 2024 ©2024 Association for Computational Linguistics

are unreliable in shorter formats. We need better metrics to make improvements on the adaptability of LLMs to education levels measurable. Only when we can measure improvements can we make tangible progress in leveraging LLMs for educational applications.[1]

As an alternative to STATIC metrics, we can use classifiers to predict the educational level of a given text. They generalize better and can be applied to texts of varying lengths. However, these classifiers are expensive to train and require more training data than we usually have for a niche domain like educational purposes. Similarly, human assessment of difficulty may provide a gold standard, but it is expensive to collect and, like all annotation tasks, suffers from disagreement.

In this paper, we introduce and evaluate a new set of PROMPT-BASED metrics for text difficulty as complements to existing STATIC metrics. PROMPT-BASED metrics are LLM prompts that exploit the general language understanding capabilities of LLMs to capture more abstract features of educational texts than STATIC metrics. For example, LLMs can flexibly classify the topic of a text, which is one adaptation technique used by teachers to adjust the content which called curriculum compacting in pedagogy (Stamps, 2004). This would be difficult to do with STATIC approaches.

We develop our selection of PROMPT-BASED metrics based on a user study, where we ask a group of university students to 1) assess the difficulty of educational texts and explain their reasoning, and 2) come up with prompts for an LLM to change the difficulty of a given text. We then translate the qualitative findings from both parts of the study into concrete LLM prompts that serve as PROMPT-BASED metrics. We incorporate prompts from other studies to manage text readability with LLMs (Imperial and Madabushi, 2023; Gobara et al., 2024). We evaluate the ability of our new PROMPT-BASED metrics to measure text appropriateness for different education levels with a series of regression experiments.

While PROMPT-BASED metrics perform on par or better than zero-shot and few-shot LLM classifiers, they are less useful for text difficulty classification by themselves than STATIC metrics. However, combining PROMPT-BASED and STATIC metrics significantly improves performance. This suggests that PROMPT-BASED metrics capture relevant signals beyond those captured by the large number of STATIC metrics.

A combination of STATIC and PROMPT-BASED metrics also provides a deeper understanding of the key metrics or features that influence complexity than classifiers could. Additionally, the factors that contribute to complexity in a scientific text differ from those in a medical or a legal document. By considering a range of metrics, we can develop more accurate domain-specific measures. Our multifaceted approach allows us to break down complexity into its basic components, such as its appropriateness for different education levels, lexical or syntactic complexity, thematic topics, and text readability.

Overall, PROMPT-BASED metrics empower educators to develop more effective content development strategies with LLMs to engage learners of all levels and backgrounds. We could have directly trained classifiers; however, this approach would not have enabled us to identify the most relevant metrics.

Contributions

1. We conduct a user study to motivate the creation of novel PROMPT-BASED metrics of text difficulty for educational texts (§2).

2. We show in a series of regression experiments that these PROMPT-BASED metrics hold additional value for text difficulty classification beyond what STATIC metrics can capture (§4.3).

3. By leveraging the interpretability of our regressions, we highlight the relative importance of individual STATIC and PROMPT-BASED metrics (§4.5).

2 User Study

Our PROMPT-BASED metrics for text difficulty are prompts based on the results of a one-day user study we ran with a group of university students in November 2023.

2.1 Study Design

The user study consisted of two main parts.

In the first part of our study, we asked participants to review 60 educational texts randomly sampled from the ScienceQA dataset (Lu et al., 2022). Each text consists of a question (e.g., “What is

[1] Similarly, metrics like BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and BLANC (Recasens and Hovy, 2011), among others, kickstarted and sustained the development of automated approaches to machine translation, summarization, and coreference resolution, respectively.

the mass of a dinner fork?") with answer choices ("70 grams or 70 kilograms") and a longer-form explanation of the solution. All texts we select here are authentic educational materials from the social, natural, or language sciences in schools. Participants were tasked with a) labeling the education level of each text as appropriate for either elementary school, middle school, or high school and b) explaining the reasoning behind their choice in a short, free-text answer.

In the second part of our study, we asked participants to rewrite scientific text explanations, also sampled from ScienceQA, to be appropriate for different education levels, with the help of an LLM – in this case, ChatGPT. For example, participants were asked to rewrite a middle school explanation of thermal energy at the elementary and high school levels with the help of prompts. We recorded the prompts they used to get ChatGPT to accomplish the adaptation for them. Thus, we collected prompts that are used both for text *simplification* and for text *complexification*.

2.2 Study Participants

We ran our study as part of a hackathon at the University of Zurich. There were seven participants aged between 21 and 31 years. Four participants were female, three male. All participants were students at Department of Computational Linguistics from University of Zurich, enrolled at the time in programs specializing in computational linguistics, computer science, and AI. Five were studying for a bachelor's degree and two for a master's degree. The participants held prior educational degrees from school systems across five different countries. Their native languages include English, Italian, German, Greek, and Ukrainian. They self-reported their English language proficiency at C1 and C2 levels. Participants were compensated in study credits that could be counted towards completing their program.

2.3 Study Results

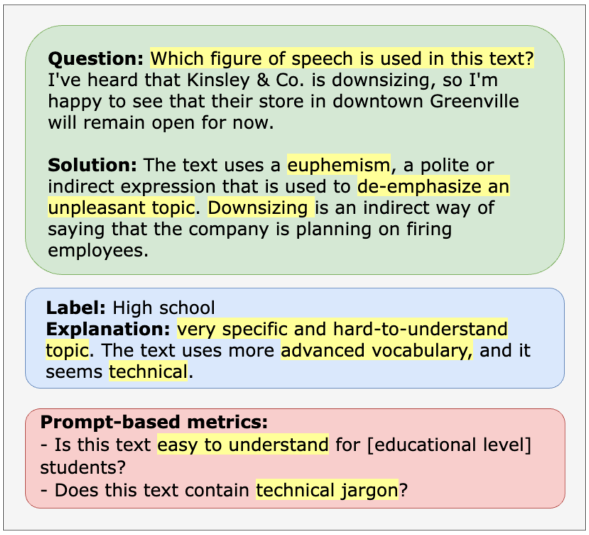
The first task of our study yielded 276 classification labels together with their corresponding descriptive justifications. These include 120 label-explanation pairs for middle school texts, 89 for high school, and 67 for elementary school texts. In the second task of our study, we collected 103 prompts for text simplification and complexification. We share illustrative examples of classifications, explanations, and prompts in Appendix A.

In the next section, we use the qualitative results from our study to motivate the construction of novel PROMPT-BASED metrics for text appropriateness for various education levels.

3 Metrics for Text Difficulty

3.1 Prompt-based Metrics

Since the metrics we introduce are based on the prompts of language models rather than discrete heuristics, we refer to them as 'PROMPT-BASED' to distinguish them. The goal of the PROMPT-BASED metrics we develop is to capture more abstract features of educational texts than would be possible with STATIC metrics, which typically focus on individual words and their statistics.



Question: Which figure of speech is used in this text?
I've heard that Kinsley & Co. is downsizing, so I'm happy to see that their store in downtown Greenville will remain open for now.

Solution: The text uses a euphemism, a polite or indirect expression that is used to de-emphasize an unpleasant topic. Downsizing is an indirect way of saying that the company is planning on firing employees.

Label: High school
Explanation: very specific and hard-to-understand topic. The text uses more advanced vocabulary, and it seems technical.

Prompt-based metrics:
- Is this text easy to understand for [educational level] students?
- Does this text contain technical jargon?

Figure 2: An illustrative example of the PROMPT-BASED metric process. The green box contains the education text from the ScienceQA dataset. The blue box shows the predicted educational level and the explanation. The red box contains the PROMPT-BASED metrics based on the sample.

We derive our PROMPT-BASED metrics from the results of our user study. Figure 2 shows an illustrative example of our derivation process. We

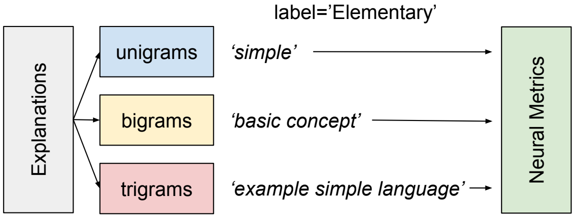


Figure 3: High-level view of the derivation process for the PROMPT-BASED metrics using n-gram frequencies. Function words are excluded.

consider users' explanations for why they consider a specific educational text to be of elementary, middle, or high school level difficulty. Then, we identify recurring attributes and other explanation features that several users mention to reflect them in PROMPT-BASED metrics. We examine the distributions of unigrams, bigrams, and trigrams across all three labels, excluding function words (see Figure 3). Some of the most frequent unigrams for the elementary level include *simple, basic, elementary*; for the high school level, *high, complex, concepts*; and for the middle school level, *explicit, explanation, middle*.

We qualitatively assessed the n-gram distributions, considering both frequencies and topic appropriateness, before finalizing the query construction. Each PROMPT-BASED metric is a simple yes-no question, which we use to prompt the LLMs. These metrics encompass the most frequent unigrams and less common bigrams and trigrams derived from the findings of our study.

While, Gobara et al. (2024) demonstrate a correlation between readability scores of LLM-generated texts in education and human assessments, Imperial and Madabushi (2023) indicate challenges in LLMs effectively adjusting the readability of text. We construct 63 PROMPT-BASED metrics using this process. Each PROMPT-BASED metric relates to either education level (30 metrics), lexical or syntactic complexity (8 metrics), and the topic of the text at hand (10 metrics). In addition, we include metrics about the text's readability score (15 metrics) based on the work by Imperial and Madabushi (2023). The complete list of all our PROMPT-BASED metrics is in Appendix C.

3.2 Existing Static Metrics

STATIC metrics are the baseline we want to improve on. All STATIC metrics are based on simple formulas, heuristics, or counts of words and other textual features. These properties make them simple to apply but limit the conceptual complexity of what they can reasonably measure. In total, we include 46 STATIC metrics, selected from those compiled in prior work (Flekova et al., 2016; Yaneva et al., 2019; Xue et al., 2020; Baldwin et al., 2021).

These metrics encompass a variety of linguistic characteristics, spanning from basic text-level measures like vocabulary size and word frequency to sentence-level attributes such as sentence length and syntactic complexity. Additionally, they take into account the question-answering structure within the input text. In the ScienceQA dataset, each question is paired with its respective solution and corresponding lecture. This segmentation of information across educational levels facilitates the computation of STATIC features for each section of the question-answer solution and lecture independently. For the complete list of 46 STATIC metrics, see Appendix C.

4 Experiments

We conduct a series of classification experiments to evaluate the usefulness of our novel PROMPT-BASED metrics for measuring text difficulty. We use a subset of the ScienceQA dataset, which contains question-answer pairs across several topics and education levels. Specifically, we run multinomial logistic regressions based on STATIC metrics, PROMPT-BASED metrics, and the combination of the two to evaluate the marginal benefits of our new PROMPT-BASED metrics. We also compare these regression approaches to using an LLM for zero-shot and few-shot classification.

4.1 Dataset

All our experiments are based on the ScienceQA dataset (Lu et al., 2022). There are 21,208 texts in ScienceQA. Each text consists of a question with answer choices, and a longer-form explanation of the solution. Texts in ScienceQA are classified according to their grade level using the K12 system from the US education system. We simplify this classification by collapsing the 12-grade levels into just three: elementary school (grades 1 to 5), middle school (grades 6 to 8), and high school (grades 9 to 12).[2] From the 21,208 texts in ScienceQA, we sample only those that do not use images in questions or explanations. We then deduplicate and sample 1,516 texts for each education level to create a balanced dataset of 4,548 texts. Of these 4,548 texts, we use 3,638 (80%) for training and 910 (20%) for evaluation. To our knowledge, ours is the first use of the ScienceQA dataset for training and evaluating text difficulty classifiers.

4.2 LLMs for Prompt-based Metrics

We use LLMs to compute the 63 PROMPT-BASED metrics described in Section 3.1. In principle, any LLM can serve this purpose. With 63 metrics for 4,548 texts, we get 286,524 prompts from each LLM. This amount is prohibitively expensive for

[2] https://usahello.org/education/children/grade-levels/

paid services like GPT4. Hence, we concentrate on state-of-the-art open LLMs, which we can execute at a low cost: Llama2 (Touvron et al., 2023), Mistral (Jiang et al., 2023), and Gemma (Google, 2024). Llama2, launched in July 2023, comprises both pre-trained and fine-tuned LLMs, ranging in size from 7 billion to 70 billion parameters. It has been reported to outperform other open-access LLMs and exhibits capabilities comparable to ChatGPT across various tasks. In this paper, we use Llama2-7b and Llama2-13b. The next model is Mistral-7B, released in September 2023, another open LLM surpassing similar-sized open LLMs. We use Mistral-7b-Instruct-v0.2, which was published in December 2023.

The last model we use is Gemma7b-it, based on the Gemma base model and trained on open-source mathematics datasets.

We set the model temperature to zero to make responses deterministic. The maximum response length is 256 tokens. Otherwise, we use standard generation parameters from the Hugging Face transformers library. We collected all responses in February 2024.

4.3 Multinomial Logistic Regression

We use simple multinomial logistic regression to classify the difficulty level of texts. The task is to predict the difficulty level C_i of a given educational text S_i. C_i can take three ordinal values: elementary, middle, or high school difficulty. Instead of including S_i directly, we include sets of STATIC and PROMPT-BASED metrics $\mathbf{M}_i$ that are computed based on S_i. We regress $\mathbf{M}_i$ on C_i on the 3,638 training texts and then evaluate on the 910 test education texts.

We vary which metrics we include across experimental setups to evaluate the marginal benefits of different metrics. There are three main setups of interest: 1) PROMPT-BASED metrics only, 2) STATIC metrics only, 3) the combination of the two, which we refer to as COMBO.

4.4 Baseline: Zero- and Few-Shot Classification

We exploit the general language capabilities of LLMs to compute PROMPT-BASED metrics, which we then use as inputs to a logistic classifier for text difficulty. A natural follow-up question is whether LLMs could directly predict text difficulty related to education levels. Therefore, we incorporate a baseline for zero-shot and few-shot text classification. We test zero-shot and few-shot classification with the same LLMs that we use for calculating our PROMPT-BASED metrics. As an additional comparison point, we test GPT-4 Turbo.

Note that while the logistic classifier is fitted to our training data, the zero-shot LLM has not seen any examples at inference time. In the few-shot setting, we provide two examples for each education level and prompt the model to assign one of the desired labels without explanations.

To investigate the effect of different prompting styles, we test five distinct prompt templates in our zero-shot setup, each consisting of 25-30 words. Additionally, each prompt contains a textual segment describing the text of the science question answering for educational-level classification. We compare performance across the five prompt templates to determine the most effective prompt, i.e., the strongest baseline for our experiments. We evaluate the models' responses on a subset of randomly selected samples (n=100). The lowest performance stands at 29%, while the highest achievement reaches 42%. We proceed with our experiments under zero-shot and few-shot setups, using the best performance style as our baselines. The selected prompt for zero-shot experiments is: "Your task is to predict the education level corresponding to a given text. You are provided with three labels to choose from: 1) elementary school 2) middle school 3) high school. Text: [text] Educational level: "

We instructed LLMs to return one of the education levels. Due to the difficulty of LLMs in directly predicting the levels and complexity of the text, we have responses without the desired educational level. In this case, we assigned a default level to this invalid response, which is the "elementary level". For example, Llama2-13b has 2.86% invalid in zero-shot and 4.07% in few-shot. The most-predicted class is elementary school level, with 75.93% in zero-shot and 80% in few-shot. The number of invalid responses for other models is available in the Appendix D.

4.5 Results

Overall Performance Table 1 reports the overall results of our different logistic classifier setups along with the ZERO-SHOT and FEW-SHOT LLM classification baselines. We use Gemma-7b, Mistral-7b, Llama2-7b, and Llama2-13b across all referenced classification methods. GPT-4 is exclusively used in the baseline due to the high cost of experiments.

The findings highlight the consistent superiority of the COMBO approach in achieving the highest macro-F1 score, surpassing all other models. Specifically, while the Llama2-7b model exhibits comparatively lower performance when employing the Prompt-based method, the Llama2-13b model demonstrates the best performance across PROMPT-BASED metrics. Notably, the Gemma-7b model stands out as the best-performing model when using the COMBO metric. In terms of Prompt-based regression, the average macro-F1 score across all models stands at 0.62, with all PROMPT-BASED metrics obtained directly through LLMs' binary classification prompts. The best performance overall is achieved by COMBO, which combines both sets of metrics, resulting in a macro-F1 score of 0.86.

Nearly all models encounter difficulty in predicting the educational level across both ZERO-SHOT and FEW-SHOT methodologies. However, in these experiments, the FEW-SHOT approach notably enhances the macro-F1 score. Additionally, Table 1 highlights that the best performance among baseline approaches is achieved by GPT-4, attaining a macro-F1 score of 0.63 in the FEW-SHOT setting.

Performance by Education Level To delve into the performance more comprehensively, we split out the results for each regression setup by label, i.e., education level, in Table 2. Here, we display only the top-performing model based on the PROMPT-BASED metric and provide the details of the other models in Appendix D.

The overall picture of PROMPT-BASED regression shows that it faces difficulty in the classification of educational level, while STATIC performs much better, and COMBO performs best, which indicates that there is an additional benefit to including the PROMPT-BASED metrics.

We collect 1,000 bootstrap samples to train and test the logistic regression models for each approach. This method helps in understanding the variability and reliability of the model performance. We use t-tests to determine if the observed differences in accuracies are statistically significant over COMBO vs. STATIC. Results in Table 2 indicate a statistically significant improvement.

Feature Importance One big benefit of our regression approach over, for example, classification with an LLM, is that we can easily measure the feature importance of each metric that goes into the classification result. For this purpose, we calculate univariate F-tests between each metric and the difficulty level variable. Table 3 shows the top-five most important features, each among the PROMPT-BASED and the STATIC metrics, based on these F-tests for *Llama2-13b* model.

Most notably, the PROMPT-BASED metrics are generally less important than the STATIC metrics. On average, the top five most important STATIC metrics are at least twice as significant as the top five PROMPT-BASED metrics. The STATIC metrics mainly focus on readability and lexical diversity, while PROMPT-BASED metrics capture topic relevancy and the inclusion of simple examples. Although they may not carry the same weight, all of the top metrics are highly statistically significant.

5 Discussion

5.1 The Value of Prompt-based Metrics

PROMPT-BASED metrics by themselves may not be a good-enough basis for classifying text difficulty (Table 1). STATIC metrics are much more effective by comparison. However, our results also show that PROMPT-BASED metrics do indeed capture relevant features of the text that are not captured by STATIC metrics since models that combine both kinds of metrics clearly perform best overall. This is despite the fact that the STATIC metrics we include are many and highly diverse.

The practical usefulness of the particular PROMPT-BASED metrics outlined in this paper is evident. Moreover, the broader application of PROMPT-BASED metrics holds promise for evaluating text complexity. Our experiments indicate that the COMBO approach outperforms other models consistently. Notably, most models exhibit superior macro-F1 scores in predicting elementary-level texts, suggesting that distinguishing science questions at the elementary level is more discernible compared to other educational levels.

Furthermore, we present the feature importance of PROMPT-BASED metrics, noting that the primary PROMPT-BASED metrics pertain to readability, understandability, and suitability of text for particular educational levels. Additionally, topic relevance (e.g., math or natural science) emerges as a significant feature. In top 5 best features of STATIC metrics are summarized through readability scores ranging from the Gunning Fog Index to the Flesch-Kincaid Index, along with a metric evaluating the lexical diversity of the text.

Method	Gemma-7b	Mistral-7b	Llama2-7b	Llama2-13b	GPT-4
Prompt-based Reg.	0.73	0.54	0.45	0.77	-
Static Reg.	0.81	**0.81**	**0.81**	0.81	-
Combo Reg.	**0.95**	**0.82**	**0.81**	**0.88**	-
Zero-shot LLM	0.35	0.34	0.35	0.35	0.51
Few-shot LLM	0.37	0.37	0.45	0.47	**0.65**

Table 1: Macro-F1 for difficulty classification on test. Prompt-based metrics, zero-shot, and few-shot (two examples) performance are specific to each LLM. Static metrics are the same across models. Zero-shot and few-shot classification use GPT4. Best performance per model in **bold**.

	Level	Precision	Recall	F1-Score
Prompt	Elem.	0.84	0.82	0.83
	Middle	0.84	0.64	0.73
	High	0.68	0.84	0.75
Static	Elem.	0.86	0.85	0.86
	Middle	0.75	0.71	0.73
	High	**0.84**	0.88	0.84
Combo	Elem.	**0.95***	**0.93***	**0.94***
	Middle	**0.89***	**0.77***	**0.83***
	High	0.82	**0.93***	**0.87***

Table 2: Difficulty classification performance on test. $*$ = statistically significant improvements of Combo over Static at $p = 0.05$ (bootstrap). Prompt-based metrics use *Llama2-13b*. Best performance per level in **bold**.

Better Prompt-based metrics identified in future work may be even more effective complements to Static metrics.

5.2 Limitations

Limited Scope of User Study The user study we conducted provides a clear empirical motivation for the Prompt-based metrics we selected. This in itself is a core contribution of our work. However, due to resource and time constraints, the sample of participants in the study is fairly small and of limited diversity. Future work could improve on our approach by conducting larger studies or recruiting participants from even more relevant professions (e.g. teachers) to motivate the selection of even better Prompt-based metrics.

Limited Availability of Relevant Data Our experiments are mostly constrained by the availability of relevant data for text difficulty classification. The ScienceQA dataset that we use is, to our knowledge, the only dataset that fits our experimental setup in terms of size and detail on education level. Therefore, we cannot make any strong claims about the generalisability of our results. Future work could invest into building new datasets and testing cross-domain performance of both Static and Prompt-based metrics, which would give useful insights into which text features are most generally indicate of text difficulty.

6 Related Work

6.1 Question Answering Datasets in Education

The review study by AlKhuzaey et al. (2023) about the literature on item difficulty classification reveals a significant shortage of publicly accessible datasets with items that are labeled according to their difficulty levels. For example, Hsu et al. (2018) gathered their dataset from national standardized entrance tests that often concentrate on the medical and language fields, annotated with the performance data of 270,000 examinees. This study includes the necessity for a publicly accessible collection of standardized datasets and the need for further exploration into alternative methods for feature elicitation and classification modeling. The lack of publicly available datasets for measuring difficulty has led researchers toward the domain of Automatic Question Generation (AQG) in recent years. Typically, questions generated by AQG tend to be more straightforward in structure and cognitive demand than questions written by humans.

Most of these automatically generated questions are basic, primarily addressing only the first level of Bloom's taxonomy, which is focused on recall (Leo et al., 2019). Another source of educational datasets is retrieved from online learning platforms or websites specific to the study's domain. An example includes the collection of 1,657

	Rank	Metric	F
Prompt Metrics	1	Based on the **ARI**, is this text suitable for ES readers?	251.77*
	2	Is this text **relevant to curriculum** topics for ES students?	249.07*
	3	Is this text about **math**?	248.17*
	4	Is this text about **natural science**?	240.07*
	5	Does this text contain **simple examples**?	235.96*
Static Metrics	1	Gunning Fog (measures **readability**)	817.86*
	2	Coleman-Liau index (measures **readability**)	785.60*
	3	Flesch-Kincaid Reading Ease (measures **readability**)	725.15*
	4	Automated Readability Index (measures **Readability**)	686.87*
	5	Number of unique Words (measures **lexical diversity**)	613.89*

Table 3: Five most important features for PROMPT-BASED and STATIC metrics in *Llama2-13b*. Feature importance is measured using univariate F-tests. Larger F indicates higher feature importance. (ES: Elementary School, ARI: Automated Readability Index) * indicates significance at >99.999% confidence.

programming problems from LeetCode[3], labeled with the number of solutions submitted and the pass rate for each problem. Additionally, fewer datasets are from domain-specific textbooks and preparation books, particularly prevalent in the language domain for their role in training students for language proficiency exams. Domain experts developed the remaining sources to meet specific study goals, and according to AlKhuzaey et al. (2023), only 7% from school or university-level assessments.

The Stanford Question Answering Dataset (SQuAD), developed by Rajpurkar et al. (2016), features 150,000 questions in the form of paragraph-answer pairs sourced from Wikipedia articles. This dataset was utilized by Bi et al. (2021) to develop and test their models for predicting the difficulty of reading comprehension questions. Lu et al. (2022) created a multimodal science question-answering datasets, which includes 21,000 English passages from school reading exams, each accompanied by four multiple-choice questions. The ScienceQA dataset provides metadata fields for each question, including extensive solutions and general explanations which made it suitable for this study (Lu et al., 2022).

6.2 Automatic Evaluation of Educational Content

The difficulty level classification of questions presented to students is crucial for facilitating more effective and efficient learning. Pérez et al. (2012) shows teachers usually fail to identify the correct difficulty level of the questions according to their students' answers and final scores. The student's perception of the difficulty also changes across grades and subjects. AlKhuzaey et al. (2023) discovers that linguistic features significantly influence the determination of question difficulty levels in educational assessments. They have explored various syntactic and semantic aspects to understand the complexity of these questions. Crossley et al. (2019) shows the value of using crowdsourcing methods to gather human assessments of text comprehension, coupled with linguistic attributes derived from advanced readability metrics. This approach aids in creating models that explain how humans understand and process text, as well as factors influencing reading speed. Crossley et al. (2023) examined the effectiveness of new readability formulas developed on the CommonLit Ease of Readability (CLEAR) corpus using more efficient sentence-embedding models and comparing them to traditional readability formulas. They did not tru LLMs directly for difficulty classification task. In their respective studies, Imperial and Madabushi (2023), Rooein et al. (2023), and Gobara et al. (2024) leverage Large Language Models (LLMs) for content generation, focusing specifically on controlling readability scores. Their research illuminates the inherent challenges and limitations encountered when attempting to effectively adapt LLMs for this purpose.

7 Conclusion

Good teachers succeed in making the material understandable for their respective audiences. This adaptation is a complex process that goes well beyond replacing individual words and phrases. How-

[3] https://leetcode.com

ever, existing STATIC metrics for text difficulty, like the Flesch-Kincaid Reading Ease score, still focus on precisely those elements. As a result, these metrics are crude and brittle, failing to adapt to new domains and working mainly on long-form documents.

Our experiments reveal the promising potential of LLMs in predicting educational difficulty through using the PROMPT-BASED metrics rather than prompting the model directly. These metrics were derived from a small-scale user study involving students. Empirically, we demonstrate that when combined with traditional static metrics, these PROMPT-BASED metrics enhance text difficulty classification.

Our study paves the way for novel applications of LLMs in educational contexts. By involving more educational stakeholders, such as teachers, we can gather more representative PROMPT-BASED metrics, facilitating future advancements in difficulty classification.

Ethical Considerations

The participants in the user study we used in our paper were student volunteers for a course on related topics. They could leave the study at any point and were compensated in course credits that could be counted towards their study program. The study was conducted in accordance with the rules of the host university and passed its ethics assessment. The risk for harm to the participants in this setting was assessed as minimal.

Acknowledgements

Donya Rooein and Dirk Hovy were supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 949944, INTEGRATOR). Paul Röttger was supported by a MUR FARE 2020 initiative under grant agreement Prot. R20YSMBZ8S (INDOMITA). They are members of the MilaNLP group and the Data and Marketing Insights Unit of the Bocconi Institute for Data Science and Analysis (BIDSA). Anastassia Shaitarova was supported by the National Centre of Competence in Research "Evolving Language", Swiss National Science Foundation (SNSF) Agreement 51NF40 180888.

References

Samah AlKhuzaey, Floriana Grasso, Terry R Payne, and Valentina Tamma. 2023. Text-based question difficulty prediction: A systematic review of automatic approaches. *International Journal of Artificial Intelligence in Education*, pages 1–53.

Peter Baldwin, Victoria Yaneva, Janet Mee, Brian E Clauser, and Le An Ha. 2021. Using natural language processing to predict item response times and improve test construction. *Journal of Educational Measurement*, 58(1):4–30.

Sheng Bi, Xiya Cheng, Yuan-Fang Li, Lizhen Qu, Shirong Shen, Guilin Qi, Lu Pan, and Yinlin Jiang. 2021. Simple or complex? complexity-controllable question generation with soft templates and deep mixture of experts model. *arXiv preprint arXiv:2110.06560*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Scott Crossley, Joon Suh Choi, Yanisa Scherber, and Mathis Lucka. 2023. Using large language models to develop readability formulas for educational settings. In *International Conference on Artificial Intelligence in Education*, pages 422–427. Springer.

Scott A Crossley, Stephen Skalicky, and Mihai Dascalu. 2019. Moving beyond classic readability formulas: New methods and new models. *Journal of Research in Reading*, 42(3-4):541–561.

Lucie Flekova, Daniel Preoţiuc-Pietro, and Lyle Ungar. 2016. Exploring stylistic variation with age and income on twitter. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 313–319.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of applied psychology*, 32(3):221.

Seiji Gobara, Hidetaka Kamigaito, and Taro Watanabe. 2024. Do llms implicitly determine the suitable text difficulty for users? *arXiv preprint arXiv:2402.14453*.

Google. 2024. Responsible Generative AI Toolk, GemmaTechnical Report. https://ai.google.dev/gemma/docs. Accessed: March 6, 2024.

Mohammad Hosseini, Catherine A Gao, David M Liebovitz, Alexandre M Carvalho, Faraz S Ahmad, Yuan Luo, Ngan MacDonald, Kristi L Holmes, and Abel Kho. 2023. An exploratory survey about using chatgpt in education, healthcare, and research. *medRxiv*, pages 2023–03.

Fu-Yuan Hsu, Hahn-Ming Lee, Tao-Hsing Chang, and Yao-Ting Sung. 2018. Automated estimation of item difficulty for multiple-choice tests: An application of word embedding techniques. *Information Processing & Management*, 54(6):969–984.

Joseph Marvin Imperial and Harish Tayyar Madabushi. 2023. Flesch or fumble? evaluating readability standard alignment of instruction-tuned language models. *arXiv preprint arXiv:2309.05454*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Jared Leo, Ghader Kurdi, Nicolas Matentzoglu, Bijan Parsia, Ulrike Sattler, Sophie Forge, Gina Donato, and Will Dowling. 2019. Ontology-based generation of medical, multi-term mcqs. *International Journal of Artificial Intelligence in Education*, 29:145–188.

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.

Pan Lu, Swaroop Mishra, Tanglin 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. *Advances in Neural Information Processing Systems*, 35:2507–2521.

OpenAI. 2023. GPT-4 Technical Report.

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.

Elena Verdú Pérez, Luisa M Regueras Santos, María Jesús Verdú Pérez, Juan Pablo de Castro Fernández, and Ricardo García Martín. 2012. Automatic classification of question difficulty level: Teachers' estimation vs. students' perception. In *2012 Frontiers in Education Conference Proceedings*, pages 1–5. IEEE.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Marta Recasens and Eduard Hovy. 2011. Blanc: Implementing the rand index for coreference evaluation. *Natural language engineering*, 17(4):485–510.

Donya Rooein, Amanda Cercas Curry, and Dirk Hovy. 2023. Know your audience: Do llms adapt to different age and education levels? *arXiv preprint arXiv:2312.02065*.

Malik Sallam. 2023. Chatgpt utility in healthcare education, research, and practice: Systematic review on the promising perspectives and valid concerns. *Healthcare*, 11(6).

Lisa S Stamps. 2004. The effectiveness of curriculum compacting in first grade classrooms. *Roeper Review*, 27(1):31–41.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Shriyash Upadhyay, Etan Ginsberg, and Chris Callison-Burch. 2023. Improving mathematics tutoring with a code scratchpad. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 20–28.

Kang Xue, Victoria Yaneva, Christopher Runyon, and Peter Baldwin. 2020. Predicting the difficulty and response time of multiple choice questions using transfer learning. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 193–197.

Lixiang Yan, Lele Sha, Linxuan Zhao, Yuheng Li, Roberto Martinez-Maldonado, Guanliang Chen, Xinyu Li, Yueqiao Jin, and Dragan Gašević. 2023. Practical and ethical challenges of large language models in education: A systematic literature review. *arXiv preprint arXiv:2303.13379*.

Victoria Yaneva, Peter Baldwin, Janet Mee, et al. 2019. Predicting the difficulty of multiple choice questions in a high-stakes medical exam. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 11–20.

A Selected Prompts from the User Study

We collect the top prompts of the students from the chat history with analytical, manual, and AI Assistant (ChatGPT).

A.1 Elementary School:

- Simplify a text for elementary school, using simple language for 6-12 years olds. - Create an elementary version of a high school lecture text. - Simplify a high school text for elementary school. - Explain in a way an 8-year-old would understand. - This is a text meant for high school students. Can you help me make an appropriate version for elementary school students with very simple language and comprehensive, easy-to-understand examples?

A.2 Middle School:

- Give examples from middle school lectures. - Adapt a high school text for middle school, using less advanced language. - Be more textbook-like and more to the point for the level of middle school. - Adapt content for a student in middle school. - Simplify a lecture text for middle school using illustrative examples.

A.3 High School:

- Enhance scientific accuracy and add comprehensive examples for the high school level. - Adapt a middle school text for high school, using advanced language. - Increase difficulty for high school, with advanced vocabulary and scientific concepts. - Can you make it more scientific and less story-telling-like? - Increase the difficulty level with comprehensive examples.

B Parameter settings

The Static metrics are collected by Python packages such as `nltk(3.8.1)`, `pandas(2.2.0)`, `textstat(0.7.3`, `spacy(3.7.4)`. We use `nltk.download` to get data for 'stopwords', 'cmudict', 'wordnet', and 'averaged_perceptron_tagger'.

We performed 8-bit quantization for collecting prompt-based metrics and baselines, with a maximum input length of 2048 tokens and a maximum output length of 256 tokens. This process was restricted to a single run due to our utilization of pre-trained models readily accessible in HuggingFace Transformers.

For Regression model, we use scikit-learn package and SelectKBest with `f_classif` score function.

C List of Metrics

C.1 Static Metrics

Table 4 shows all static metrics.

C.2 Prompt-based Metrics

Is this text readable for an elementary school student?, Is this text suitable for an elementary school student?, Is this text easy to understand for elementary school students?, Is this text relevant to curriculum topics for elementary school students?, Is this text relevant to the knowledge and experiences of elementary school students?, Could an average elementary school student engage with the content of this task?, Could most elementary school students complete this task without significant difficulty?, Is this text appropriate for the skills and knowledges of elementary school students?, Is the length of this text suitable for elementary school students?, Would the vocabulary in this text be comprehensible to elementary school students?, Is this text readable for a middle school student?, Is this text suitable for a middle school student?, Is this text easy to understand for middle school students?, Is this text relevant to curriculum topics for middle school students?, Is this text relevant to the knowledge and experiences of middle school students?, Could an average middle school student engage with the content of this task?, Could most middle school students complete this task without significant difficulty?, Is this text appropriate for the skills and knowledges of middle school students?, Is the length of this text suitable for middle school students?, Would the vocabulary in this text be comprehensible to middle school students?, Is this text readable for a high school student?, Is this text suitable for a high school student?, Is this text easy to understand for high school students?, Is this text relevant to curriculum topics for high school students?, Is this text relevant to the knowledge and experiences of high school students?, Could an average high school school student engage with the content of this task?, Could most high school students complete this task without significant difficulty?, Is this text appropriate for the skills and knowledges of high school students?, Is the length of this text suitable for high school students?, Would the vocabulary in this text be comprehensible to high school students?, Does this text contain metaphors and/or figurative language?, Does this text use complex language?, Does this text use simple language?, Does this text contain technical jargon?, Is this text about science?, Is this text about language science?, Is this text about natural science?, Is this text about social science?, Is this text about math?, Is this text about physics?, Is this text about chemistry?, Is this text about earth science?, Is this text about world history?, Is this text about geography?, Based on the Flesch-Kincaid reading-ease score, is this text suitable for elementary school readers?, Based on the Flesch-Kincaid reading-ease score, is this text suitable for middle school readers?, Based on the Flesch-Kincaid reading-ease score, is this text suitable for high school readers?, Based on the Gunning Fog Index, is this text suitable for elementary school readers?, Based on the Gunning Fog Index, is this text suitable for middle school readers?, Based on the Gunning Fog Index, is this text suitable for high school readers?, Based on the Coleman-Liau Index, is this text suitable for elementary school readers?, Based on the Coleman-Liau Index, is this text suitable for middle school readers?, Based on the Coleman-Liau Index, is this text suitable for high school readers?, Based on the Automated Readability Index (ARI), is this text suitable for elementary school readers?, Based on the Automated Readability Index (ARI), is this text

Table 4: List of Static metrics

Feature	Description
n_words_q	Number of words in the question
n_words_a_solution	Number of words in the solution of an answer
n_words_a_lecture	Number of words in the lecture
Text_Length	Length of the text
Word_Count	Total word count
Nouns	Number of nouns
Verbs	Number of verbs
Adjectives	Number of adjectives
Adverbs	Number of adverbs
Num_Numbers	Number of numeric characters
Num_Commas	Number of commas
Num_Complex_Words	Number of complex words
Num_Unique_Words	Number of unique words
Num_Content_Words	Number of content words
Num_Content_Words_No_Stopwords	Number of content words excluding stopwords
Word_Length_Syllables	Average word length in syllables
Avg_Sentence_Length	Average sentence length
Num_Prepositional_Phrases	Number of prepositional phrases
Num_Negated_Words_Stem	Number of negated words stemmed
Num_Negated_Words_Lead_In	Number of negated words leading in
Num_Main_Noun_Phrases	Number of main noun phrases
Avg_Main_NP_Length	Average length of main noun phrases
Num_Verb_Phrases	Number of verb phrases
Prop_Active_Voice_Verbs	Proportion of active voice verbs
Prop_Passive_Voice_Verbs	Proportion of passive voice verbs
Ratio_Active_to_Passive_Verbs	Ratio of active to passive voice verbs
Num_Words_Before_Main_Verb	Number of words before the main verb
Num_Agentless_Passive_Constructions	Number of agentless passive constructions
Word_Length_Std_Dev	Standard deviation of word lengths
Num_Polysemic_Words	Number of polysemic words
Num_Word_Senses	Number of word senses
Num_Word_Senses_For_Content_Words	Number of word senses for content words
Num_Word_Senses_For_Nouns	Number of word senses for nouns
Num_Word_Senses_For_Verbs	Number of word senses for verbs
Num_Word_Senses_For_Non_Auxiliary_Verbs	Number of word senses for non-auxiliary verbs
Num_Word_Senses_For_Adjectives	Number of word senses for adjectives
Num_Word_Senses_For_Adverbs	Number of word senses for adverbs
Distance_To_Root_Nouns	Distance to root for nouns
Distance_To_Root_Verbs	Distance to root for verbs
flesch_kincaid_grade	Flesch-Kincaid grade level
flesch_kincaid_ease	Flesch-Kincaid ease score
coleman_liau_index	Coleman-Liau index
automated_readability_index	Automated Readability Index
smog_index	SMOG index
gunning_fog	Gunning Fog index
traenkle_bailer_index	Traenkle-Bailer index

suitable for middle school readers?, Based on the Automated Readability Index (ARI), is this text suitable for high school readers?, Based on the SMOG Index, is this text suitable for elementary school readers?, Based on the SMOG Index, is this text suitable for middle school readers?, Based on the SMOG Index, is this text suitable for high school readers?, Does this text contain basic concepts that are easy to comprehend?, Does this text cover multiple concepts?, Does this text provide a very explicit explanation?, Does this text contain simple examples?

D Details over Gemma-7B, Mistral-7B, and Llama2-7B

We describe the performance of these models in detail. Gemma7b has 10.33% invalid response in zero-shot and 9.56% over few-shot. The majority of the predicted class is high school level 73.41% in zero-shot and 72.75% in few-shot. Mistral7b has 15.49% invalid response in zero-shot and 6.37% invalid in few-shot and with majority of classification for high school level with 66.04% in zero-shot and 42.31% for elemetary school in few-shot. Llama2-7b has 13.08% invalid in zero-shot and 5.49% in few-shot and the majority of elementary school classification with 66.26% in zero-shot and also 76.04% in few-shot. Gpt-4 has only 5.93% invalid in zero-shot and 0.77% in few-shot. Gpt-4 predicted also the high school level as the highest classification with 41.54% in zero-shot and 40.22% in few-shot.

	Level	Precision	Recall	F1-Score
PROMPT-BASED	Elem.	0.83	0.81	0.82
	Middle	0.75	0.57	0.65
	High	0.66	0.81	0.65
STATIC	Elem.	0.86	0.85	0.86
	Middle	0.75	0.71	0.73
	High	0.84	0.88	0.86
COMBO	Elem.	**0.98***	**0.98***	**0.98***
	Middle	**0.98***	**0.91***	**0.95***
	High	**0.91***	**0.97***	**0.94***

Table 5: Difficulty classification performance on test. $*$ = statistically significant improvements of COMBO over STATIC at $p = 0.05$ (bootstrap). PROMPT-BASED metrics use *Gemma-7b*. Best performance per level in **bold**.

	Level	Precision	Recall	F1-Score
PROMPT	Elem.	0.46	0.86	0.60
	Middle	**0.92**	0.83	**0.88**
	High	0.34	0.10	0.16
STATIC	Elem.	**0.86**	0.85	0.86
	Middle	0.75	0.71	0.73
	High	0.84	**0.88**	**0.86**
COMBO	Elem.	0.76	**0.95***	**0.84***
	Middle	0.85	**0.90***	**0.88***
	High	**0.89** *	0.64	0.75

Table 6: Difficulty classification performance on test. $*$ = statistically significant improvements of COMBO over STATIC at $p = 0.05$ (bootstrap). PROMPT-BASED metrics use *Mistral-7b*. Best performance per level in **bold**.

	Level	Precision	Recall	F1-Score
PROMPT	Elem.	0.44	0.47	0.45
	Middle	0.62	0.61	0.62
	High	0.29	0.28	0.28
STATIC	Elem.	0.86	0.85	0.86
	Middle	**0.75**	0.71	**0.73**
	High	**0.84**	**0.88**	**0.86**
COMBO	Elem.	**0.88***	**0.97***	**0.93***
	Middle	0.72	**0.74***	**0.73***
	High	0.83	0.73	0.78

Table 7: Difficulty classification performance on test. $*$ = statistically significant improvements of COMBO over STATIC at $p = 0.05$ (bootstrap). PROMPT-BASED metrics use *Llama2-7b*. Best performance per level in **bold**.

	Rank	Metric	F
Prompt	1	Based on the **Coleman-Liau Index**, is the text suitable for MS readers?	105.09*
	2	Is this text **readable** for a MS student?	104.42*
	3	Based on the **SMOG Index**, is this text suitable for MS readers?	103.53*
	4	Is this text **suitable** for a MS student?	94.21*
	5	Based on the **Gunning Fog Index**, is this text suitable for MS readers?	92.35*
Static Metrics	1	Gunning Fog (measures text **readability**)	817.86*
	2	Coleman-Liau index (measures text **readability**)	785.60*
	3	Flesch-Kincaid Reading Ease (measures **readability**)	725.15*
	4	Automated Readability Index (measures **lexical diversity**)	686.87*
	5	Number of unique Words (measures **lexical diversity**)	613.89*

Table 8: Top five most important features among the PROMPT-BASED and STATIC metrics. Feature importance is measured using univariate F-tests. Larger F indicates higher feature importance. (MS: Middle School) PROMPT-BASED metrics use the *Gemma-7B* model. * indicates significance at >99.999% confidence.

	Rank	Metric	F
Prompt	1	Based on the **Gunning Fog Index**, is this text suitable for ES readers?	209.84*
	2	Is this text **easy to understand** for ES students??	193.22*
	3	Is this text **Suitable** for ES students	190.61*
	4	Is this text about **math**?	175.72*
	5	Is this text **relevant to curriculum** topics for ES students?	175.08*
Static Metrics	1	Gunning Fog (measures text **readability**)	817.86*
	2	Coleman-Liau index (measures text **readability**)	785.60*
	3	Flesch-Kincaid Reading Ease (measures **readability**)	725.15*
	4	Automated Readability Index (measures **Readability**)	686.87*
	5	Number of unique Words (measures **lexical diversity**)	613.89*

Table 9: Top five most important features among the PROMPT-BASED and STATIC metrics. Feature importance is measured using univariate F-tests. Larger F indicates higher feature importance. (ES: Elementary School) PROMPT-BASED metrics use the *Mistral-7B* model. * indicates significance at >99.999% confidence.

	Rank	Metric	F
Prompt-based Metrics	1	Is this text **relevant to curriculum** topics for ES students?	139.66*
	2	Is this text suitable for an ES student?	136.97*
	3	Is this text **readable** for an ES student	132.89*
	4	Based on the **Gunning Fog Index**, is this text **suitable** for MS readers?"	125.51*
	5	Is this text about **natural science**?	124.52*
Static Metrics	1	Gunning Fog (measures text **readability**)	817.86*
	2	Coleman-Liau index (measures text **readability**)	785.60*
	3	Flesch-Kincaid Reading Ease (measures **readability**)	725.15*
	4	Automated Readability Index (measures **Readability**)	686.87*
	5	Number of unique Words (measures **lexical diversity**)	613.89*

Table 10: Top five most important features among the PROMPT-BASED and STATIC metrics. Feature importance is measured using univariate F-tests. Larger F indicates higher feature importance.(ES: Elementary School, MS: Middle School) PROMPT-BASED metrics use the *Llamma2-7B* model. * indicates significance at >99.999% confidence.

Large Language Models Are State-of-the-Art Evaluator for Grammatical Error Correction

Masamune Kobayashi[◇] **Masato Mita**[†◇] **Mamoru Komachi**[‡]
[◇]Tokyo Metropolitan University, Japan [†]CyberAgent Inc. [‡]Hitotsubashi University, Japan
kobayashi-masamune@ed.tmu.ac.jp, mita_masato@cyberagent.co.jp,
mamoru.komachi@r.hit-u.ac.jp

Abstract

Large Language Models (LLMs) have been reported to outperform existing automatic evaluation metrics in some tasks, such as text summarization and machine translation. However, there has been a lack of research on LLMs as evaluators in grammatical error correction (GEC). In this study, we investigate the performance of LLMs in English GEC evaluation by employing prompts designed to incorporate various evaluation criteria inspired by previous research. Our extensive experimental results demonstrate that GPT-4 achieved Kendall's rank correlation of 0.662 with human evaluations, surpassing all existing methods. Furthermore, in recent GEC evaluations, we have underscored the significance of the LLMs scale and particularly emphasized the importance of fluency among evaluation criteria.

1 Introduction

Large Language Models (LLMs) have surpassed existing systems in various NLP tasks, showcasing their high capabilities of language understanding and generation (Ye et al., 2023; Bubeck et al., 2023). These LLMs, which have had a significant impact on recent NLP research, also demonstrate the ability to produce high-quality corrections in grammatical error correction (GEC) (Schick et al., 2022; Dwivedi-Yu et al., 2022; Fang et al., 2023; Loem et al., 2023; Coyne et al., 2023).

In recent years, several studies have been conducted on the use of LLMs as an evaluator. In text summarization, dialogue generation, and machine translation, GPT-4 has demonstrated superior performance compared to existing automatic evaluation metrics (Liu et al., 2023b; Kocmi and Federmann, 2023). While there is very little research on GEC evaluation, considering GPT-4's ability to explain grammatical errors with 90% accuracy in human evaluations (Song et al., 2023), it holds potential for evaluating corrections. Sottana et al. (2023) conducted meta-evaluation using a limited number of systems, but there has been no comprehensive analysis using dozens of systems like traditional approaches such as Grundkiewicz et al. (2015) and Kobayashi et al. (2024).

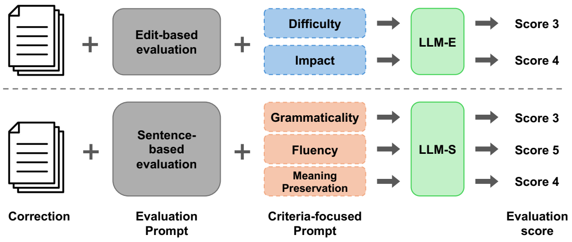


Figure 1: Evaluation framework using LLMs.

Therefore, we aim to explore the extent to which LLMs operate as evaluation models in English GEC. Specifically, we conduct GEC evaluations using LLMs with prompts at different evaluation granularities to investigate how evaluation capabilities change with the presence of evaluation criteria and the scale of LLMs, as shown in Figure 1. Kobayashi et al. (2024)'s work on the evaluation of metrics (i.e., meta-evaluation) has revealed that conventional metrics lack the resolution to capture performance differences in high-performing GEC systems. Given this current state, to facilitate proper GEC evaluation moving forward, we investigate the potential of LLMs by comparing them with conventional metrics through meta-evaluation.

Our contributions are summarized as follows. (1) We conducted a comprehensive investigation into the performance of LLMs as evaluators in GEC, and the results showed that GPT-4 achieved state-of-the-art performance, indicating the usefulness of considering evaluation criteria in prompts (especially fluency). (2) It was suggested that as LLM scales decrease, the correlation with human evaluations decreases, and the ability to capture fluency in corrected sentences diminishes. Smaller LLMs tend to avoid extreme scores, while larger LLMs

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 68–77
June 20, 2024 ©2024 Association for Computational Linguistics

tend to assign higher scores.

2 Experiment setup

In this section, we explain the considered GEC metrics (§2.1) and meta-evaluation methods (§2.2).

2.1 Considered metrics

GEC metrics: We use two types of evaluation metrics: Edit-Based Metrics (EBMs), which assess only the edits made in the corrected text, and Sentence-Based Metric (SBMs), which evaluate the overall quality of the corrected sentences.

For EBMs, we employ four metrics.

- M^2 **(Dahlmeier and Ng, 2012)** dynamically extracts edits using Levenshtein algorithm to maximize overlap with gold annotations from the hypothesis sentences and calculates the F-score.

- **ERRANT (Bryant et al., 2017)** is similar to M^2, but it differs in that it uses a linguistically extended Damerau-Levenshtein algorithm for edit extraction to enhance the alignment of tokens with similar linguistic properties.

- **GoToScorer (Gotou et al., 2020)** calculates an F-score taking into account the difficulty of corrections. The difficulty is defined based on the number of systems that could correctly correct errors per total number of systems.

- **PT-M^2 (Gong et al., 2022)** combines M^2 with BERTScore (Zhang et al., 2019), enabling the measurement of semantic similarity in addition to simply comparing edits.

For SBMs, we utilize four metrics.

- **GLEU (Napoles et al., 2015)** rewards n-grams in the hypothesis sentence that match the reference but are not in the source sentence while penalizing n-grams in the source that do not match the reference. We use GLEU without tuning (Napoles et al., 2016).

- **Scribendi Score (Islam and Magnani, 2021)** evaluates based on GPT-2 perplexity, token sort ratio, and Levenshtein distance ratio.

- **SOME (Yoshimura et al., 2020)** fine-tunes BERT (Devlin et al., 2019) using human evaluation scores based on three criteria: grammaticality, fluency, and meaning preservation.

- **IMPARA (Maeda et al., 2022)** utilizes a quality estimation model and a similarity model based on BERT to consider the impact of edits.

LLMs: We consider three LLMs: LLaMa 2 (Touvron et al., 2023) (13B for chat), GPT-3.5 (Ouyang et al., 2022) (`gpt-3.5-turbo-1106`), and GPT-4 (OpenAI, 2023) (`gpt-4-1106-preview`), conducting evaluations using prompts to assess both edits and sentences separately. LLMs for edit-based evaluation are denoted with "-E" at the end, while ones for sentence-based evaluation have "-S" at the end. Furthermore, we created prompts focusing on GEC evaluation criteria to investigate the impact of prompts on evaluation performance, comparing them with the base prompt. For simplicity, this experiment uses only GPT-4 as the base LLM architecture. GPT-4-E, which evaluates edits, focuses on the difficulty of corrections (Gotou et al., 2020) and the impact of edits (Maeda et al., 2022). GPT-4-S, which evaluates sentences, uses prompts focusing on grammaticality, fluency, and meaning preservation (Asano et al., 2017; Yoshimura et al., 2020). Detailed information on each prompt is provided in Appendix A.

2.2 Meta-evaluation methods

We conduct system-level and sentence-level meta-evaluations using SEEDA dataset (Kobayashi et al., 2024). SEEDA consists of human evaluations at two different granularities: edit-based and sentence-based, for 12 outputs from neural-based GEC systems and 3 human-authored sentences. The dataset comprises two components: SEEDA-E based on edit-based evaluation and SEEDA-S based on sentence-based evaluation. In SEEDA, for correction pairs (A, B) sampled from these corrected sentence collections, three annotators provide 5-point scores for each granularity, resulting in 5347 pairwise judgments (A>B, A=B, A<B). Subsequently, human rankings (from 1st to 15th place) of systems are obtained from pairwise judgments using rating algorithms such as Trueskill (Sakaguchi et al., 2014) and Expected Wins (Bojar et al., 2013). We conduct two variations of meta-evaluation: "Base", which uses the 12 systems excluding outliers, and "+ Fluent corr.", which adds two fluent corrected sentences[1] additionally.

[1] In GEC, there are two types of edits: minimal edits, which make the minimum necessary corrections, and fluency edits, which aim to make the sentence more fluent.

Metric	System-level								Sentence-level							
	SEEDA-E				SEEDA-S				SEEDA-E				SEEDA-S			
	Base		+ Fluent corr.		Base		+ Fluent corr.		Base		+ Fluent corr.		Base		+ Fluent corr.	
	r	ρ	r	ρ	r	ρ	r	ρ	Acc	τ	Acc	τ	Acc	τ	Acc	τ
M^2	0.791	0.764	-0.239	0.161	0.658	0.487	-0.336	-0.013	0.582	0.328	0.527	0.216	0.512	0.200	0.496	0.170
ERRANT	0.697	0.671	-0.502	0.051	0.557	0.406	-0.587	-0.116	0.573	0.310	0.511	0.188	0.498	0.189	0.471	0.129
GoToScorer	0.901	0.937	0.667	0.916	0.929	0.881	0.627	0.881	0.521	0.042	0.505	0.009	0.477	-0.046	0.504	0.009
PT-M^2	0.896	0.909	-0.083	0.442	0.845	0.769	-0.162	0.336	0.587	0.293	0.542	0.200	0.527	0.204	0.528	0.180
GLEU	0.911	0.897	0.053	0.482	0.847	0.886	-0.039	0.475	0.695	0.404	0.630	0.266	0.673	0.351	0.611	0.227
Scribendi Score	0.830	0.848	0.721	0.847	0.631	0.641	0.611	0.717	0.377	-0.196	0.359	-0.240	0.354	-0.238	0.345	-0.264
SOME	0.901	0.951	0.943	0.969	0.892	0.867	0.931	0.916	0.747	0.512	0.743	0.494	0.768	0.555	0.760	0.531
IMPARA	0.889	0.944	0.935	0.965	0.911	0.874	0.932	0.921	0.742	0.502	0.725	0.455	0.761	0.540	0.742	0.496
GPT-3.5-E	-0.059	0.182	-0.844	-0.257	-0.270	-0.245	-0.900	-0.525	0.463	-0.073	0.428	-0.143	0.487	-0.026	0.437	-0.126
GPT-4-E	0.911	0.965	0.845	0.974	0.839	0.846	0.786	0.899	0.728	0.455	0.702	0.404	0.698	0.395	0.687	0.374
+ Difficulty	0.941	0.972	0.909	0.978	0.885	0.860	0.863	0.908	0.719	0.437	0.708	0.417	0.717	0.434	0.703	0.406
+ Impact	0.905	**0.986**	0.848	**0.987**	0.844	0.860	0.793	0.908	0.730	0.460	0.710	0.420	0.717	0.434	0.696	0.392
Llama 2-S	0.534	0.427	0.161	0.349	0.482	0.273	0.090	0.235	0.521	0.042	0.527	0.054	0.534	0.068	0.526	0.052
GPT-3.5-S	0.878	0.916	0.302	0.648	0.770	0.636	0.199	0.433	0.633	0.265	0.597	0.195	0.631	0.263	0.608	0.216
GPT-4-S	0.960	0.958	0.967	0.969	0.887	0.860	0.931	0.908	0.798	0.595	0.783	0.565	0.784	0.567	0.770	0.540
+ Grammaticality	0.961	0.937	0.981	0.956	0.888	0.867	**0.953**	0.912	0.807	0.615	0.804	0.607	0.796	0.592	0.788	0.577
+ Fluency	**0.974**	0.979	**0.981**	0.982	0.913	0.874	0.952	0.916	**0.831**	**0.662**	**0.812**	**0.624**	**0.819**	**0.637**	**0.797**	**0.594**
+ Meaning Preservation	0.911	0.960	0.976	0.974	**0.958**	**0.881**	0.952	**0.925**	0.813	0.626	0.793	0.587	0.810	0.620	0.792	0.584

Table 1: Results of system-level and sentence-level meta-evaluations. GPT-4-S demonstrated higher performance compared to existing GEC metrics, showing the most improvement in correlation when focusing on fluency.

System-level meta-evaluation: In the system-level meta-evaluation, we utilize the system scores derived from human rankings of systems using TrueSkill (Sakaguchi et al., 2014). For metrics like SOME, where system-level scores cannot be directly calculated, we use the average of sentence-level scores as a substitute. Additionally, for LLMs, we employ system scores derived from LLMs rankings (Appendix B) similar to human rankings. To measure the correlation between human evaluations and metric scores, we use Pearson correlation (r) and Spearman rank correlation (ρ). To ensure proper correlation calculation, we use the set of sentences that humans evaluated to compute the metric scores.

Sentence-level meta-evaluation: In the sentence-level meta-evaluation, we use pairwise judgments from SEEDA. To investigate the proximity between human evaluations and metric scores, we employ Accuracy (Acc) and Kendall's rank correlation (τ). Kendall (τ) is valuable for assessing performance in common use cases where corrections are compared to each other.

3 Results

In this section, we analyze the performance of LLMs as GEC evaluators in system-level (§3.1) and sentence-level meta-evaluations (§3.2). Additionally, we conduct further analysis by changing the system set to investigate the impact of the considered systems in the meta-evaluation(§3.3).

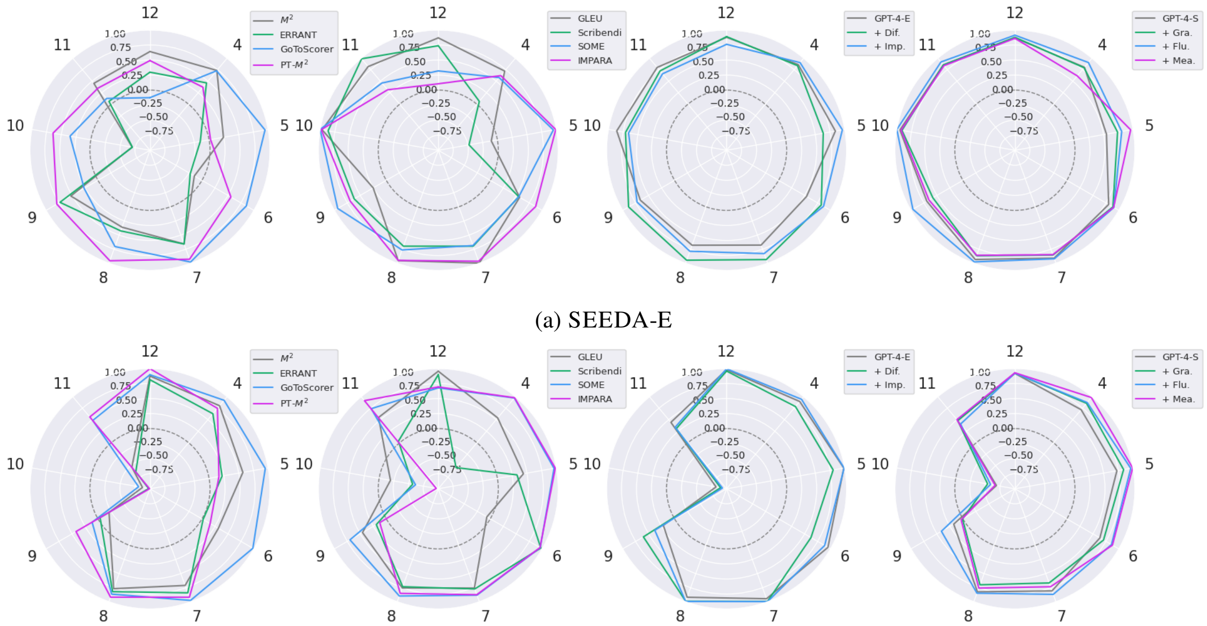
3.1 System-level analysis

In Table 1,[2] GPT-4 tends to achieve high correlations compared to existing metrics, highlighting their utility in GEC evaluations. These prompts that focus on criteria tend to enhance correlation compared to base prompts, implying that GPT-4 can derive valuable insights from evaluation criteria. This observation aligns with recent studies that report performance improvements by incorporating additional sentences into the prompt (Barham et al., 2022; Kojima et al., 2023; Li et al., 2023).

The decrease in correlation as the LLM scale decreases, such as with Llama 2 and GPT-3.5, suggests the importance of the LLM scale. Especially, the decrease in correlation when adding fluent corrected sentences ("+ Fluent corr.") compared to "Base" implies that smaller-scale LLMs may not adequately consider the fluency of sentences. Possible reasons for this include issues such as LLM's tendency to produce the same scores (Appendix C) and the inability to interpret the context of prompts as expected by users. However, GPT-4 consistently demonstrated a high correlation and provided more stable evaluations compared to traditional metrics.

The fact that most system-level correlations for GPT-4 exceed 0.9 suggests that the conventional meta-evaluation using a dozen systems may have reached a performance saturation point for the task. This poses a significant concern as it could lead to an underestimation of high-performing metrics in future meta-evaluations. One possible solution is to utilize sentence-level correlations with a larger

[2] Llama 2-E was excluded from this experiment because its output scores were not stable.



(a) SEEDA-E

(b) SEEDA-S

Figure 2: Window analysis was performed by selecting any consecutive four systems from the human rankings of the 12 systems ("Base"). For instance, x=4 involves calculating the Pearson correlation (r) using the systems ranked from 1st to 4th in the human rankings. In contrast to conventional GEC metrics, which exhibit unstable correlations, GPT-4 demonstrates relatively stable correlations.

sample size or explore correlations between systems with similar performance levels, increasing the difficulty of the task.

3.2 Sentence-level analysis

In the sentence-level meta-evaluation, we observed differences in correlations between metrics that were not apparent in the system-level meta-evaluation. In particular, while GPT-4-E and GPT-4-S showed similar correlations in system-level meta-evaluation, it was revealed that there was a notable difference between them. Additionally, considering fluent corrected sentences ("+ Fluent corr.") led to a slight decrease in overall correlation, but GPT-4 still maintained a considerably high correlation compared to traditional metrics. This suggests that GPT-4 exhibits strong correlations with human evaluations and that examining sentence-level correlations is beneficial for comparing high-performance metrics.

Most prompts focused on criteria significantly improved sentence-level correlations compared to the base prompt. Notably, GPT-4-S + Fluency demonstrated the ability to greatly enhance performance, surpassing existing GEC metrics and achieving state-of-the-art performance. This suggests the need for a detailed examination of fluency beyond grammaticality when evaluating high-quality corrections. Paradoxically, it implies that humans also prioritize fluency when comparing high-quality corrected sentences. Furthermore, the moderate fluctuations in correlation resulting from changing a single word in the prompt (GPT-4-S + Grammaticality vs. GPT-4-S + Fluency) highlight the impact of prompt engineering on performance. In other aspects, the results were generally consistent with those in the system-level meta-evaluation.

3.3 Further analysis

To increase the difficulty of the meta-evaluation task, we computed correlations using a set of systems with similar performance. Specifically, we conduct system-level meta-evaluation using only subsets of consecutive four systems in the human rankings of systems, and show the transitions of correlation at positions from 4th to 12th as window analysis in Figure 2[3]. For example, the point at x=4 represents the Pearson correlation value calculated using only the outputs of the four systems ranked from 1st to 4th.

According to the window analysis, GPT-4 maintains relatively high and stable correlations, making

[3] For simplicity, we exclude the results of Llama 2 and GPT-3.5, which showed low performance.

them suitable for evaluating modern neural systems in recent years. In SEEDA-E, the notably high correlations of GPT-4-S + Fluency across almost all data points emphasize the importance of fluency. In SEEDA-S, while overall correlations are high, the significant decrease in correlation at x=10, suggests the presence of GEC systems that are challenging to evaluate for the metrics. On the other hand, conventional metrics frequently exhibit either no correlation or negative correlation, indicating their low robustness in GEC evaluation.

4 Related Work

Several studies have investigated the evaluation performance of LLMs. Chiang and Lee (2023) conducted the first investigation into LLM evaluation performance, demonstrating that GPT-3.5 can achieve expert-level evaluation in tasks such as open-ended story generation and adversarial attacks. In the summarization task, Liu et al. (2023a) revealed that GPT-4 has state-of-the-art evaluation performance by leveraging their proposed methods like auto-CoT (Chain-of-Thought) and weighted scores. In the machine translation task, Kocmi and Federmann (2023) demonstrated that only larger models exceeding GPT-3.5 can perform translation quality evaluation, with GPT-4 slightly inferior to existing metrics at the segment level. Yancey et al. (2023) utilized LLMs to evaluate second language writing proficiency through essay grading, discovering that GPT-4 exhibits performance equivalent to modern automated writing evaluation methods.

5 Conclusion

In this work, we investigated the capability of LLMs as evaluators in English GEC, and GPT-4 demonstrated significantly higher correlations compared to traditional metrics. Future work should delve into the impact of few-shot learning and optimize prompt engineering for enhanced evaluation performance. Furthermore, we plan to explore the possibility of document-level evaluation, considering the expansion of the GPT's context window, which is not currently focused on by existing metrics.

6 Limitations

Some of the LLMs (such as GPT-4) used in this study are not freely available and may require special access or payment to use. This could limit the applicability of our evaluation method. Additionally, since many LLMs are constantly updated, there is a possibility of inconsistent evaluation results across different versions. To address this issue, we also conducted evaluations using reproducible LLMs (such as Llama 2).

Acknowledgments

This work is partly supported by JST, PRESTO Grant Number JPMJPR2366, Japan.

References

Hiroki Asano, Tomoya Mizumoto, and Kentaro Inui. 2017. Reference-based metrics can be replaced with reference-less metrics in evaluating grammatical error correction systems. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 343–348, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Paul Barham, Aakanksha Chowdhery, Jeff Dean, Sanjay Ghemawat, Steven Hand, Dan Hurt, Michael Isard, Hyeontaek Lim, Ruoming Pang, Sudip Roy, Brennan Saeta, Parker Schuh, Ryan Sepassi, Laurent El Shafey, Chandramohan A. Thekkath, and Yonghui Wu. 2022. Pathways: Asynchronous distributed dataflow for ML.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria. 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.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with GPT-4.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.

Steven Coyne, Keisuke Sakaguchi, Diana Galvan-Sosa, Michael Zock, and Kentaro Inui. 2023. Analyzing the performance of GPT-3.5 and GPT-4 in grammatical error correction.

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.

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.

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.

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.

Peiyuan Gong, Xuebo Liu, Heyan Huang, and Min Zhang. 2022. Revisiting grammatical error correction evaluation and beyond. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6891–6902, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Takumi Gotou, Ryo Nagata, Masato Mita, and Kazuaki Hanawa. 2020. Taking the correction difficulty into account in grammatical error correction evaluation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2085–2095, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Edward Gillian. 2015. Human evaluation of grammatical error correction systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 461–470, Lisbon, Portugal. Association for Computational Linguistics.

Md Asadul Islam and Enrico Magnani. 2021. Is this the end of the gold standard? a straightforward reference-less grammatical error correction metric. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3009–3015, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Masamune Kobayashi, Masato Mita, and Mamoru Komachi. 2024. Revisiting meta-evaluation for grammatical error correction.

Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 193–203, Tampere, Finland. European Association for Machine Translation.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners.

Cheng Li, Jindong Wang, Yixuan Zhang, Kaijie Zhu, Wenxin Hou, Jianxun Lian, Fang Luo, Qiang Yang, and Xing Xie. 2023. Large language models understand and can be enhanced by emotional stimuli.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023a. G-eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Yixin Liu, Alexander R. Fabbri, Jiawen Chen, Yilun Zhao, Simeng Han, Shafiq Joty, Pengfei Liu, Dragomir Radev, Chien-Sheng Wu, and Arman Cohan. 2023b. Benchmarking generation and evaluation capabilities of large language models for instruction controllable summarization.

Mengsay Loem, Masahiro Kaneko, Sho Takase, and Naoaki Okazaki. 2023. Exploring effectiveness of GPT-3 in grammatical error correction: A study on performance and controllability in prompt-based methods. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 205–219, Toronto, Canada. 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.

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, Matt Post, and Joel Tetreault. 2016. Gleu without tuning.

OpenAI. 2023. GPT-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll 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 F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme. 2014. Efficient elicitation of annotations for human evaluation of machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 1–11, Baltimore, Maryland, USA. Association for Computational Linguistics.

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.

Yixiao Song, Kalpesh Krishna, Rajesh Bhatt, Kevin Gimpel, and Mohit Iyyer. 2023. Gee! grammar error explanation with large language models.

Andrea Sottana, Bin Liang, Kai Zou, and Zheng Yuan. 2023. Evaluation metrics in the era of GPT-4: Reliably evaluating large language models on sequence to sequence tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8776–8788, Singapore. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Kevin P. Yancey, Geoffrey Laflair, Anthony Verardi, and Jill Burstein. 2023. Rating short L2 essays on the CEFR scale with GPT-4. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 576–584, Toronto, Canada. Association for Computational Linguistics.

Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, Jie Zhou, Siming Chen, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. A comprehensive capability analysis of GPT-3 and GPT-3.5 series models.

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.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating text generation with BERT. *CoRR*, abs/1904.09675.

A Prompts for GEC evaluation

The prompts used for edit-based evaluation and sentence-based evaluation by LLMs are illustrated in Figures 3a and 3b, respectively. In the # context, [SOURCE] represents the source, and [PREVIOUS] and [FOLLOWING] are the preceding and succeeding sentences in the essay, respectively. In the # targets, [CORRECTION N WITH EDITS] denotes a corrected sentence with explicitly indicated edits, while [CORRECTION N] represents a regular corrected sentence. Here, N takes values from 1 to 5. Additionally, the prompts output scores in JSON format to maintain a consistent output format. For prompts focused on evaluation criteria, the following sentence is added to the end of the first paragraph of the prompt.

- Difficulty: "Please evaluate each edit in the target with a focus on the difficulty of corrections."
- Impact: "Please evaluate each edit in the target with a focus on its impact on the sentence."
- Grammaticality: "Please evaluate each target with a focus on the grammaticality of the sentence."
- Fluency: "Please evaluate each target with a focus on the fluency of the sentence."
- Meaning Preservation: "Please evaluate each target with a focus on preserving the meaning between each target and the source, which is the middle sentence in the context."

An example of a prompt for evaluation using GPT-4-S + Fluency is provided below:

The goal of this task is to rank the presented targets based on the quality of the sentences.

The context consists of three sentences from an essay written by an English learner.

After reading the context to understand the flow, please assign a score from a minimum of 1 point to a maximum of 5 points to each target based on the quality of the sentence (note that you can assign the same score multiple times).

Please evaluate each target with a focus on the fluency of the sentence.

# context

These are the advantages that save works most of the time .

In conclude , socia media benefits people in several ways but in the same time harms people .

People should avoid the misuse of socia media and use it in the proper way .

# targets

In conclude , socia media benefits people in several ways but in the same time harms people .

In conclusion , social media benefits people in several ways but at the same time harms people .

In conclusion , social media benefits people in several ways but , at the same time , harms people .

In conclude , social media benefits people in several ways but at the same time harms people .

In conclusion , socia media benefits people in several ways but , at the same time , harms people .

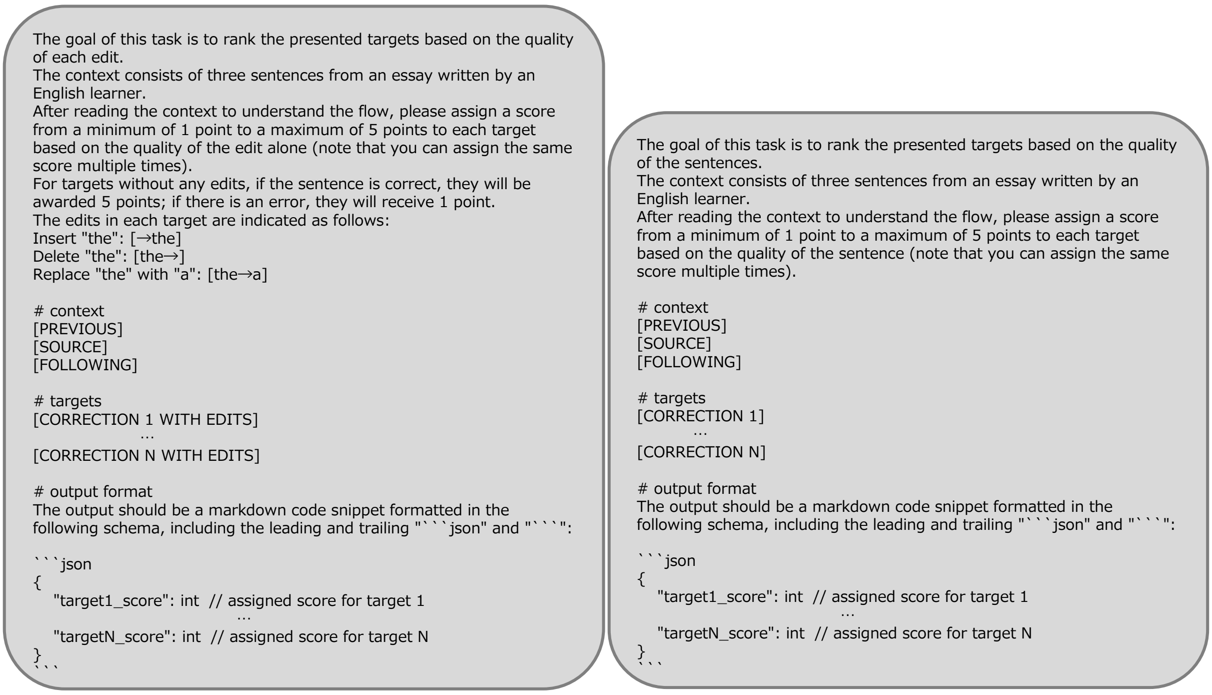
# output format ...

B LLM rankings of GEC systems

The LLM rankings based on pairwise judgments (A>B, A=B, A<B) of corrections (A, B) conducted by LLMs and generated using Trueskill are shown in Table 2. It can be observed that LLMs with relatively smaller scales, such as GPT-3.5 and Llama2, have difficulty in ranking fluent corrections (REF-F and GPT-3.5) higher. Furthermore, these LLMs tend to assign similar scores to many systems, suggesting that they may not effectively differentiate between the quality of corrections. In contrast, GPT-4 can rank fluent corrections highly, resulting in rankings that closely resemble human evaluations.

C Tendency of LLM scoring

The distribution of scores assigned by LLMs to corrected sentences is shown in Figure 4. As the LLM scale increases, there is a tendency to assign higher scores (4 or 5 points). Based on our meta-evaluation results, which suggest that higher LLM scales are associated with higher correlations with human evaluations, smaller LLMs may underestimate corrections judged to be good by humans. Llama 2-S tends to avoid extreme scores such as 1 or 5 points and shows a high degree of score overlap, making it difficult to compare more detailed corrected sentences.



~~~
The goal of this task is to rank the presented targets based on the quality
of each edit.
The context consists of three sentences from an essay written by an
English learner.
After reading the context to understand the flow, please assign a score
from a minimum of 1 point to a maximum of 5 points to each target
based on the quality of the edit alone (note that you can assign the same
score multiple times).
For targets without any edits, if the sentence is correct, they will be
awarded 5 points; if there is an error, they will receive 1 point.
The edits in each target are indicated as follows:
Insert "the": [→the]
Delete "the": [the→]
Replace "the" with "a": [the→a]

# context
[PREVIOUS]
[SOURCE]
[FOLLOWING]

# targets
[CORRECTION 1 WITH EDITS]
          ...
[CORRECTION N WITH EDITS]

# output format
The output should be a markdown code snippet formatted in the
following schema, including the leading and trailing "```json" and "```":

```json
{
   "target1_score": int  // assigned score for target 1
                                          ...
   "targetN_score": int  // assigned score for target N
}
```
~~~

(a) Edit-based evaluation

~~~
The goal of this task is to rank the presented targets based on the quality
of the sentences.
The context consists of three sentences from an essay written by an
English learner.
After reading the context to understand the flow, please assign a score
from a minimum of 1 point to a maximum of 5 points to each target
based on the quality of the sentence (note that you can assign the same
score multiple times).

# context
[PREVIOUS]
[SOURCE]
[FOLLOWING]

# targets
[CORRECTION 1]
          ...
[CORRECTION N]

# output format
The output should be a markdown code snippet formatted in the
following schema, including the leading and trailing "```json" and "```":

```json
{
   "target1_score": int  // assigned score for target 1
                                          ...
   "targetN_score": int  // assigned score for target N
}
```
~~~

(b) Sentence-based evaluation

Figure 3: Prompts used for edit-based evaluation and sentence-based evaluation by LLMs

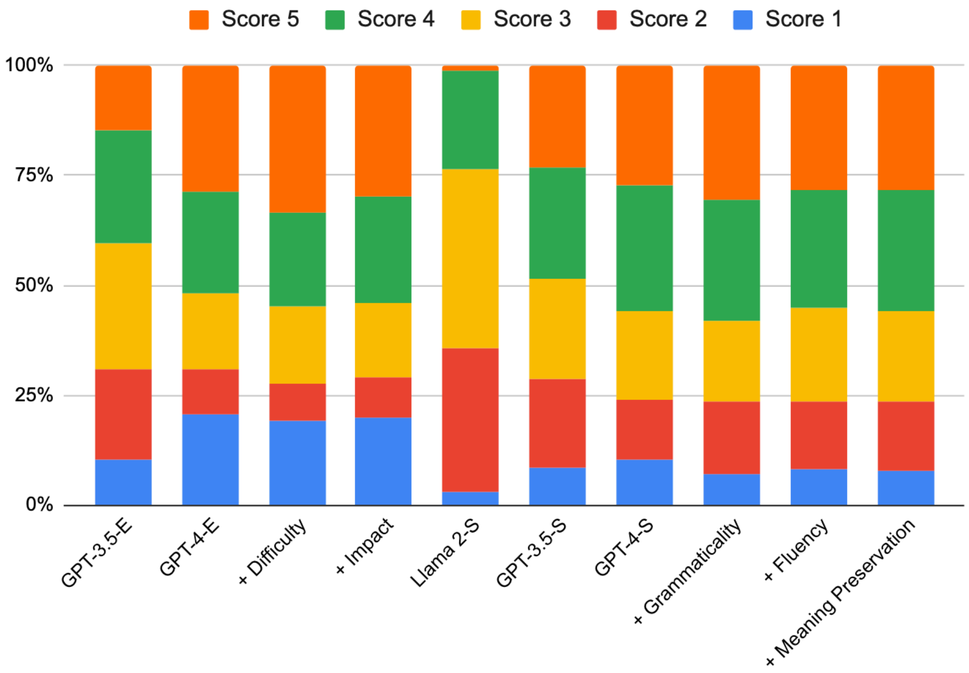


Figure 4: The distribution of scores assigned by LLMs on a 5-point scale. It can be observed that as the LLM scale increases, there is a tendency to assign higher scores (4 or 5 points). Based on our meta-evaluation results indicating better correlation with human judgments as the scale increases, it is suggested that smaller LLMs may underestimate corrections judged to be good by humans.

#	Score	Range	System
1	0.481	1	INPUT
2	0.287	2	UEDIN-MS
3	0.215	3-3	GECToR-ens
4	0.110	4-6	Riken-Tohoku
	0.089	4-8	GECToR-BERT
	0.078	4-8	TransGEC
	0.066	4-9	PIE
	0.032	6-12	REF-M
	0.025	7-12	BERT-fuse
	0.017	7-13	LM-Critic
	-0.005	8-13	BART
	-0.008	8-13	T5
	-0.011	9-13	TemplateGEC
5	-0.460	14	GPT-3.5
6	-0.916	15	REF-F

(a) GPT-3.5-E

#	Score	Range	System
1	0.409	1	GPT-3.5
2	0.210	2-4	REF-F
	0.182	2-4	TransGEC
	0.148	3-6	T5
	0.127	3-7	REF-M
	0.105	4-8	BERT-fuse
	0.075	6-9	UEDIN-MS
	0.071	6-9	Riken-Tohoku
	0.064	6-9	GECToR-BERT
	0.003	9-11	PIE
	-0.06	10-11	LM-Critic
3	-0.147	12-13	TemplateGEC
	-0.150	12-13	GECToR-ens
4	-0.266	14	BART
5	-0.770	15	INPUT

(b) GPT-4-E

#	Score	Range	System
1	0.440	1	GPT-3.5
2	0.304	2	REF-F
3	0.186	3-5	TransGEC
	0.169	3-5	T5
	0.134	4-7	BERT-fuse
	0.102	5-8	Riken-Tohoku
	0.095	5-8	REF-M
	0.054	7-9	UEDIN-MS
	0.021	8-10	PIE
	-0.007	9-10	GECToR-BERT
4	-0.138	11-13	LM-Critic
	-0.145	11-13	GECToR-ens
	-0.179	11-14	TemplateGEC
	-0.227	13-14	BART
5	-0.809	15	INPUT

(c) GPT-4-E + Difficulty

#	Score	Range	System
1	0.429	1	GPT-3.5
2	0.237	2-4	REF-F
	0.198	2-4	TransGEC
	0.167	3-5	T5
	0.118	4-8	REF-M
	0.107	4-8	BERT-fuse
	0.093	5-9	Riken-Tohoku
	0.075	6-10	UEDIN-MS
	0.064	6-10	GECToR-BERT
	0.026	8-10	PIE
3	-0.129	11-13	LM-Critic
	-0.130	11-13	GECToR-ens
	-0.163	11-13	TemplateGEC
4	-0.293	14	BART
5	-0.798	15	INPUT

(d) GPT-4-E + Impact

#	Score	Range	System
1	0.104	1-4	PIE
	0.094	1-5	REF-M
	0.084	1-7	GPT-3.5
	0.058	2-7	BERT-fuse
	0.052	2-8	GECToR-ens
	0.042	3-8	TransGEC
	0.019	4-10	UEDIN-MS
	0.010	5-11	Riken-Tohoku
	-0.017	7-11	GECToR-BERT
	-0.019	7-11	T5
	-0.034	8-12	INPUT
	-0.087	10-15	REF-F
	-0.099	12-15	BART
	-0.102	12-15	TemplateGEC
	-0.104	12-15	LM-Critic

(e) Llama 2-S

#	Score	Range	System
1	0.236	1	TransGEC
2	0.170	2-5	T5
	0.143	2-6	UEDIN-MS
	0.141	2-6	REF-M
	0.116	2-7	GPT-3.5
	0.095	4-7	Riken-Tohoku
	0.048	6-9	GECToR-BERT
	0.038	6-9	BERT-fuse
	-0.004	8-10	PIE
	-0.044	9-11	GECToR-ens
	-0.080	10-13	REF-F
	-0.093	10-13	LM-Critic
	-0.141	12-14	BART
	-0.165	13-14	TemplateGEC
3	-0.458	15	INPUT

(f) GPT-3.5-S

#	Score	Range	System
1	0.658	1	GPT-3.5
2	0.542	2	REF-F
3	0.203	3-4	TransGEC
	0.187	3-5	T5
	0.145	4-6	BERT-fuse
	0.091	6-7	Riken-Tohoku
	0.074	6-7	REF-M
4	0.009	8-9	UEDIN-MS
	-0.032	8-10	GECToR-BERT
	-0.085	9-11	PIE
	-0.102	10-11	LM-Critic
5	-0.238	12-14	TemplateGEC
	-0.258	12-14	GECToR-ens
	-0.293	13-14	BART
6	-0.901	15	INPUT

(g) GPT-4-S

#	Score	Range	System
1	0.673	1-2	GPT-3.5
	0.636	1-2	REF-F
2	0.194	3-4	TransGEC
	0.184	3-4	T5
3	0.121	5-7	BERT-fuse
	0.090	5-7	Riken-Tohoku
	0.082	5-7	REF-M
	0.022	7-8	UEDIN-MS
4	-0.074	9-11	LM-Critic
	-0.076	9-11	GECToR-BERT
	-0.118	9-11	PIE
5	-0.213	12-13	TemplateGEC
	-0.238	12-13	GECToR-ens
6	-0.309	14	BART
7	-0.974	15	INPUT

(h) GPT-4-S + Grammaticality

#	Score	Range	System
1	0.721	1	GPT-3.5
2	0.648	2	REF-F
3	0.230	3-4	TransGEC
	0.178	3-5	T5
	0.122	4-6	BERT-fuse
	0.115	5-7	REF-M
	0.063	6-7	Riken-Tohoku
4	-0.007	8-9	UEDIN-MS
	-0.058	8-11	PIE
	-0.066	9-11	GECToR-BERT
	-0.102	9-11	LM-Critic
5	-0.264	12-14	GECToR-ens
	-0.271	12-14	TemplateGEC
	-0.308	12-14	BART
6	-1.002	15	INPUT

(i) GPT-4-S + Fluency

#	Score	Range	System
1	0.653	1-2	REF-F
	0.601	1-2	GPT-3.5
2	0.242	3-4	T5
	0.209	3-4	TransGEC
3	0.135	5-6	REF-M
	0.106	5-7	BERT-fuse
	0.071	6-7	Riken-Tohoku
4	0.011	8	UEDIN-MS
5	-0.067	9-10	GECToR-BERT
	-0.106	9-11	LM-Critic
	-0.123	10-11	PIE
	-0.225	12-13	TemplateGEC
	-0.255	12-13	GECToR-ens
7	-0.317	14	BART
8	-0.935	15	INPUT

(j) GPT-4-S + Meaning Preservation

Table 2: LLM rankings generated using Trueskill based on pairwise judgments made by LLMs. GPT-4 ranks fluent corrections (REF-F, GPT-3.5) highly, resulting in these rankings that closely resemble human ranking.

Can Language Models Guess Your Identity? Analyzing Demographic Biases in AI Essay Scoring

Alexander Kwako and **Christopher Ormerod**
Cambium Assessment Inc.
alexander.kwako@cambiumassessment.com
christopher.ormerod@cambiumassessment.com

Abstract

Large language models (LLMs) are increasingly used for automated scoring of student essays. However, these models may perpetuate societal biases if not carefully monitored. This study analyzes potential biases in an LLM (XLNet) trained to score persuasive student essays, based on data from the PERSUADE corpus. XLNet achieved strong performance based on quadratic weighted kappa, standardized mean difference, and exact agreement with human scores. Using available metadata, we performed analyses of scoring differences across gender, race/ethnicity, English language learning status, socioeconomic status, and disability status. Automated scores exhibited small magnifications of marginal differences in human scoring, favoring female students over males and White students over Black students. To further probe potential biases, we found that separate XLNet classifiers and XLNet hidden states weakly predicted demographic membership. Overall, results reinforce the need for continued fairness analyses as use of LLMs expands in education.

1 Introduction

As Large Language Models (LLM)s are increasingly used for Automated Essay Scoring (AES), it is crucial that we thoroughly analyze these systems for biases (Rodriguez et al., 2019). Given that LLMs are pretrained on large corpora, they have the potential to inherit biases embedded in the functions that predict word probabilities (Bhardwaj et al., 2021). If the potential biases are not monitored carefully with fairness in mind, they risk perpetuating and amplifying existing societal biases against vulnerable populations. Rigorous demographic analysis of AES systems help ensure they live up to principles of equity and fairness.

The Persuasive Essays for Rating, Selecting, and Understanding Argumentative and Discourse Elements (PERSUADE) corpus provides a valuable resource for analyzing bias in AES system (Crossley et al., 2022). PERSUADE contains over twenty-five thousand persuasive student essays that were annotated for argumentative elements in addition to holistic grades assigned by human raters. What makes this corpus ideal for the analysis of bias is the rich metadata about the students including gender, race, and other demographic indicators. This allows for in-depth analysis of an automated scoring system's performance on essays written by students of diverse demographic affiliations.

Our goal is to investigate potential biases in LLMs trained using conventional techniques that aim to replicate human-assigned holistic scores. After training the LLM scoring model, we evaluate whether or not automated scores introduce (or exacerbate) biases relative to human-assigned scores (Ormerod et al., 2022). After evaluating bias, we determine whether the set of features that the LLM uses for scoring also contains information relevant to demographic membership. In other words, can the LLM guess certain demographic characteristics based on the scoring model? Linear modeling using these features was recently used as evidence of model validity in AES (Ormerod, 2022). The novelty of this study lies in applying these techniques and showing their relevance to the analyses of bias in LLMs.

Broadly, our research aims are as follows:

1. Fine-tune an LLM to score students' essays and assess model performance.

2. Evaluate the fine-tuned LLM for biases relative to human-assigned scores, based on students' demographic affiliations.

3. Determine whether demographic affiliation can be predicted by the (hidden layer of the) fine-tuned LLM. As a helpful reference point, assess whether separate LLMs can be fine-tuned to predict demographic affiliation.

These aims help us determine if LLMs are able to score students' essays fairly, and if demographic affiliations are an implicit feature of the scoring model.

2 Methods

2.1 Data

The PERSUADE dataset consists of a 25,488 essay responses to 15 prompts written by students from Grades 6 to 12.[1] Each essay was assigned a holistic essay score according to a rubric available with the dataset.

The prompts were administered to students within specific grades or grade-bands. For comparability with other studies, we used the same train-test split as was used in the original Kaggle competition; we created a development dataset (or *dev set*) from a random subset of the training data, for use in model selection, early stopping, and hyperparameter optimization. Table 1 shows sample sizes of train-dev-test splits, along with the average word count, for each prompt.

Demographic data was included for all prompts, but not all prompts included every demographic characteristic. For most prompts, however, we analyzed potential biases of the following demographic affiliations:

- Gender: M = Male and F = Female
- Race/ethnicity: W = White, L = Hispanic/Latino, B = Black/African American, A = Asian/Pacific Islander
- English Language Learners: ELL = Identified as an English language learner.
- Economically Disadvantaged: SES = Identified as economically disadvantaged, based on eligibility for K-12 federal assistance programs.
- Disability Status: DS = Identified as having a disability. Type of disability unspecified.

2.2 Scoring Model

One of the problems in the application of conventional pretrained LLMs, such as BERT (Devlin et al., 2019), is that the transformer architecture imposes a fixed context length (Vaswani et al., 2017; Mayfield and Black, 2020). There is an extensive body of literature that has addressed this length limitation, e.g. Longformer (Beltagy et al., 2020), Transformer-XL (Dai et al., 2019), and XLNet (Yang et al., 2019). These innovations are particularly suited for AES systems, which require longer context lengths.

Among the longer-context models, XLNet performs particularly well on AES and argumentation annotation (Ormerod et al., 2023). The key feature of XLNet is its recurrent form of attention (Dai et al., 2019).

Automated scoring generally benefits from using a regression head (with MSE loss) as opposed to a classification head (with cross-entropy loss) since regression parsimoniously retains the ordinal nature of score points (Ormerod et al., 2021).

We used the Adam optimizer with a weight decay mechanism (Loshchilov and Hutter, 2019). The learning rate was set to 5×10^{-6} with a linear learning rate scheduler, in batches of 8. Models were trained over 20 epochs, with early stopping determined by best performance on the dev set. To prevent out of memory errors, max token length was set to 2,048.

2.3 Performance Metrics

We assess the system's performance using the three standard metrics proposed by Williamson et al. (2012) for the evaluation of automated scoring systems. These include Cohen's quadratic weighted kappa (QWK, Cohen, 1960), standardized mean difference (SMD), and exact agreement.

These agreement statistics quantify the proximity of automated scores to human-assigned scores. Most operational standards consider model performance relative to human-human levels of agreement; however, only final score was included in the corpus. Nevertheless, Crossley et al. (2022) report that all essays were scored independently by two human raters and, across all PERSUADE items, the QWK was .745. Item-specific QWKs were not reported. SMD was also not reported. In the absence of double-scored data, a QWK of at least 0.7 and an SMD of at most 0.15 are commonly-accepted guidelines for adequate performance.

2.4 Analytic Approach Toward Bias

There are marginal (i.e. first-order) differences in score point distributions and in expected scores between demographic groups (Appendix A). For instance, female students generally score higher than male students on persuasive writing. It is pos-

[1] The PERSUADE corpus is publicly available online at https://github.com/scrosseye/persuade_corpus_2.0.

Prompt	Prompt Name	Grade	N_{Train}	N_{Dev}	N_{Test}
1	Phones and driving	N/A	558	140	464
2	Exploring Venus	10	740	185	923
3	Community service	8	608	153	773
4	Seeking multiple opinions	8	1232	309	7
5	Facial action coding system	10	880	221	1062
6	Distance learning	9-12	1192	299	656
7	Summer projects	9-12	696	175	872
8	Cell phones at school	8	663	166	824
9	Car-free cities	10	784	197	973
10	Grades for extracurricular activities	8	648	163	808
11	The face on Mars	8	654	164	764
12	Does the electoral college work?	9	1448	362	228
13	Driverless cars	10	1098	275	496
14	Mandatory extracurricular activities	8	668	167	824
15	"A Cowboy Who Rode the Waves"	6	546	137	682
Overall		**6-12**	**12422**	**3106**	**10356**

Table 1: A summary of how the data was split for training purposes.

sible that these group differences reflect biases in human-assigned scores; however, it is also possible that these group differences reflect legitimate differences in writing proficiency. Without additional information (e.g. a set of "unbiased" items, as would be used in an analysis of differential item functioning), the source of these differences cannot be determined.

The ambiguity of interpreting group differences extends to interpreting differences between automated and human-assigned scores. In *absolute* terms, for instance, differences could indicate that LLMs are introducing biases or, on the contrary, eliminating biases. As such, we limit ourselves to making claims in *relative* terms, i.e., do LLMs introduce biases relative to human scores?

2.5 Matching

On average, some groups scored higher or lower than others (e.g. female students scored higher than males, on average). To adjust for these marginal differences, we compared male and female students who received 1s to each other, male and female students who received 2s, etc., which is known as *exact matching* (Ho et al., 2011). Exact matching is ideal in this research context given that our sample is large, leaving very few students unmatched, even within specific prompts. As opposed to literally matching one student with another, we employ exact matching to produce a set of sample weights which, when taken as a whole, eliminate marginal group differences. These sample weights are used in subsequent analyses.

2.6 Group Difference Estimation

To compute human-XLNet scoring differences (i.e., relative bias), we estimated pairwise group differences. Regression estimates were produced using cluster-robust standard errors (Bell and McCaffrey, 2002; Pustejovsky and Tipton, 2018), as implemented by Blair et al. (2024) in R 4.3.1 (R Core Team, 2023). We used exact matching weights, described above, in these analyses.

2.7 Controlling False Discovery Rate

To avoid making spurious claims that are a product of random chance, we controlled the false discovery rate using the Benjamini-Hochberg (B-H) technique (Benjamini and Hochberg, 1995). We use the term *statistically significant* when an estimated p-value is below the B-H adjusted p-value. In practical terms, B-H adjusted p-values 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. 49).

2.8 Predicting Demographic Affiliation

We predict demographic affiliation using two complementary methods. The first, more conventional method, is to train separate XLNet models to classify students' demographic affiliation based on their text responses. For example, we trained one model to predict gender, another model to predict race / ethnicity, etc.

The second method of predicting demographic affiliation was to use the hidden state from the scoring model for predictions. That is, for each demographic characteristic, linear models were trained using the hidden state as features.[2] More technically, the XLNet model used for scoring, $\mathcal{M}$, is a function of the input text, x, and can be broken into five distinct components:

$$\mathcal{M}(x) = \underbrace{(\sigma \circ \mathcal{L})}_{\text{Classifier}} \circ \underbrace{(\mathcal{S} \circ \mathcal{T} \circ \mathcal{E})}_{\substack{\text{feature}\\ \text{model}}}(x) \tag{1}$$

where $\mathcal{E}$ is the embedding, $\mathcal{T}$ is the function for the layers of (recurrent) transformers, $\mathcal{S}$ is a summary layer that extracts the information for classification, $\mathcal{L}$ is a linear layer, and σ is the activation function. Conceptually, these five components can be grouped into a *feature model* and a *classifier*. The feature model maps text to a vector space of features that are subsequently used by the linear classifier to determine the score.

In predicting demographic characteristics, we used the following model:

$$\tilde{\mathcal{M}}(x) = (\sigma \circ \tilde{\mathcal{L}}) \circ (\mathcal{S} \circ \mathcal{T} \circ \mathcal{E})(x) \tag{2}$$

Here, the feature model is frozen and $\tilde{\mathcal{L}}$ is optimized to predict demographic affiliation. If $\tilde{\mathcal{M}}$ can accurately distinguish demographic affiliation, using the language of Ormerod (2022), we say that the feature is *implicit* in the model. For example, in the ASAP dataset (Shermis, 2014), Ormerod (2022) demonstrated that essay length was an implicit feature of the model because it was a linear combination of the scoring features.

3 Results

We organize our findings around three foci. First, we evaluate the performance of XLNet to ensure it meets operational standards. Second, we assess the fairness of XLNet's automated scores by determining if there are any discrepancies, based on students' demographic affiliations, as compared to human-assigned scores. Finally, we determine the extent to which the scoring model has demographic features embedded within it.

3.1 Model Performance

We determined model performance on a prompt-by-prompt basis, as well as aggregated over all prompts. Table 2 summarizes the performance of the model in terms of three common agreement statistics: quadratic weighted kappa (QWK), standardized mean difference (SMD), and accuracy (all of which are described in greater detail in section 2.3).

Prompt	QWK	SMD	Acc	N
1	0.781	-0.066	0.683	464
2	0.856	0.003	0.677	923
3	0.800	-0.109	0.693	773
4	0.674	-0.312	0.429	7
5	0.865	-0.116	0.696	1062
6	0.875	0.042	0.697	656
7	0.813	-0.051	0.634	872
8	0.800	-0.021	0.717	824
9	0.796	-0.087	0.616	973
10	0.779	-0.025	0.699	808
11	0.818	0.063	0.658	764
12	0.863	-0.011	0.649	228
13	0.774	0.215	0.621	496
14	0.815	0.163	0.659	824
15	0.755	-0.040	0.691	682
Overall	0.864	-0.010	0.672	10356

Table 2: The performance of the model trained to the holistic scores in terms of the agreement with the human assigned scores.

Based on commonly-accepted operational standards, three items are in violation of these standards. More specifically, Prompts 4, 13, and 14 have high SMDs. Results for one of these items (Prompt 4), however, is unreliable due to the small test set sample size. Overall, however, XLNet performs well; indeed, in terms of overall QWK, XLNet exceeds human-human reliability.

3.2 Automated Scoring Biases

To measure automated scoring biases, we estimated pairwise differences between reference and focal groups. Table 3 displays the results of our automated scoring bias analysis, with standard errors in

[2] To clarify, XLNet (with a regression head) was first fine-tuned to predict score; after fine-tuning, we replaced the regression head with a classification head, froze all other layers, and fine-tuned again (using the same hyperparameters) to predict demographic characteristics.

Prompt	**F-M**	**B-W**	**L-W**	**A-W**	**SES**	**ELL**	**DS**
1	0.07 (0.06)	0.03 (0.04)	0.09 (0.05)	0.31 (0.22)			
2	0.10 (0.05)	0.00 (0.04)	-0.01 (0.05)	-0.06 (0.07)	-0.04 (0.05)	-0.10 (0.07)	-0.19 (0.02)
3	0.09 (0.04)	-0.15 (0.09)	-0.17 (0.03)	0.19 (0.09)	-0.11 (0.04)	-0.15 (0.07)	-0.35 (0.05)
5	0.07 (0.03)	-0.12 (0.02)	-0.12 (0.04)	0.01 (0.09)	-0.06 (0.01)	-0.09 (0.04)	-0.08 (0.03)
6	0.05 (0.02)	0.03 (0.10)	-0.08 (0.07)	0.07 (0.13)	-0.15 (0.08)	-0.28 (0.10)	-0.07 (0.08)
7	0.04 (0.04)	-0.29 (0.06)	-0.12 (0.04)	0.10 (0.04)	-0.06 (0.02)	-0.13 (0.09)	-0.12 (0.06)
8	0.09 (0.04)	-0.19 (0.07)	-0.11 (0.02)	0.14 (0.16)	-0.12 (0.03)	-0.19 (0.03)	-0.18 (0.10)
9	0.12 (0.02)	-0.13 (0.05)	-0.06 (0.05)	0.16 (0.17)		-0.16 (0.08)	
10	0.13 (0.03)	-0.20 (0.09)	-0.14 (0.06)	-0.04 (0.06)	-0.19 (0.05)	-0.26 (0.11)	0.03 (0.11)
11	0.09 (0.06)	-0.08 (0.04)	-0.02 (0.01)	-0.06 (0.05)	-0.12 (0.07)	-0.33 (0.09)	-0.11 (0.06)
12	0.07 (0.08)						
13	0.14 (0.04)	-0.08 (0.03)	-0.02 (0.04)	0.05 (0.17)	-0.27 (0.05)	-0.37 (0.33)	0.04 (0.19)
14	0.12 (0.06)	-0.12 (0.06)	-0.09 (0.02)	0.04 (0.01)	-0.17 (0.03)	-0.15 (0.05)	-0.09 (0.05)
15	0.04 (0.03)	-0.08 (0.07)	0.00 (0.08)	0.11 (0.10)	-0.13 (0.07)	-0.31 (0.09)	0.08 (0.14)
Overall	**0.06 (0.01)**	**-0.07 (0.01)**	-0.06 (0.02)	0.07 (0.02)	-0.10 (0.02)	-0.10 (0.04)	-0.07 (0.02)

Table 3: Biases in XLNet scores, relative to human-assigned scores. Pairwise group differences are presented as z-scores. Bold font indicates statistically significant differences.

parentheses. Score differences were normalized so that units are in standard deviations (i.e. they may be interpreted as z scores). More specifically, a difference of 0 indicates that there was no difference between focal and reference groups; a negative difference indicates that the focal group received a lower score, on average, compared to the reference group; and a positive difference indicates that the focal group received a higher score. Differences that were statistically significant are presented in bold.

Group differences varied across prompts, but trends were generally consistent. We found no statistically significant group differences within specifics prompts.

Overall, however, we found that XLNet gave higher scores to female students compared to male students ($z = 0.06$, $SE = 0.01$, $p = .0012$), and lower scores to Black students compared to White students ($z = -0.07$, $SE = 0.01$, $p = .0023$). These differences are consistent with marginal differences observed between these groups, based on human-rater scores (Table 5). That is, XLNet magnified marginal between-group differences; the effect size, however, was small. Students with low SES status and English Language Learner status also scored lower than their respective reference groups; these differences, however, were not statistically significant.

3.3 Model-Embedded Demographics

To determine if demographic information was embedded within the scoring model, we predicted demographic affiliation from the hidden state of the model. The right side of Table 4 ("Score Features") presents the results of these analyses, with QWK (or κ) as the effect size.

According to McHugh (2012), a κ value within the range of $0 \leq \kappa \leq 0.2$ is considered to have "no agreement," $0.2 < \kappa \leq 0.4$ is considered "minimal," $0.4 \leq \kappa \leq 0.6$ is "moderate," $0.6 < \kappa \leq 0.8$ is "substantial," and anything above 0.8 is "almost perfect."

For nearly all prompts, effect sizes range from "no agreement" to "minimal agreement." The one exception is predicting ELL status in Prompt 6 ($\kappa = 0.75$), which is a substantial effect size. This suggests that XLNet was able to distinguish ELL status quite well based on students' essay responses for this prompt.

In interpreting these results, it is important to bear in mind that we have not controlled for marginal differences in students' scores or factors associated with students' scores. Some of these additional factors are listed in Appendix A. For example, length is associated with students' scores and it is well-documented that female students tend to write more than males. When essay length is used to predict gender, the strength of the relationship is $\kappa = 0.058$. Note that this effect size is only slightly better than randomly guessing the gender of the student. Using the average word count, word-length, number of sentences, and Flesch–Kincaid as features to determine gender, we obtained a κ statistic of 0.106, and $\kappa < 0.06$ for all races / ethnicities, disability status, and ELL status.

We not only predicted demographic affiliation from the scoring model, but also trained separate XLNet models to predict demographic affiliation directly from students' essays. The left side of Table 4 ("Text") presents these results. These results serve as a useful comparison, since they serve as an upper-bound of how well XLNet can predict student groups based on essay responses. κ values seem particularly high for SES and ELL.

4 Discussion

4.1 Conclusions

This study makes an important contribution to the growing body of research on bias in AES systems based on LLMs. Although XLNet generally demonstrated strong performance on key metrics compared to human raters, it also magnified marginal differences between groups, relative to human-assigned scores. In particular, relative to human-assigned scores, XLNet was found to be more generous to female students compared to male students and White students compared to Black students. Additionally, we found evidence that these group differences were embedded in the hidden layer of the model.

Although effect sizes of biases were small, in large-scale assessments even small differences can affect many students. Furthermore, in high stake settings (e.g. high-school exit exams), such differences can result in failure to meet graduation requirements. XLNet magnified marginal differences, a finding consistent with other research (Kwako et al., 2023); this indicates that marginalized populations may be particularly at risk of unfair scoring.

Overall, this study demonstrates the importance and feasibility of comprehensive bias evaluations when deploying AI scoring in high-stakes educational settings. Responsible use of automated systems requires evidence that they do not create or worsen inequities for marginalized student populations. With careful design and monitoring, LLMs should help make writing assessment more consistent, reliable, and constructive for all students.

4.2 Limitations

As stated above (Section 2.4), our claims are limited to evaluating biases relative to human scores. Yet human scores themselves are often biased (e.g. Zechner, 2019). Thus, it is possible that XLNet is more fair than human raters, in spite of it magnifying marginal group differences relative to human raters. Differential item functioning (DIF, Angoff, 1993) accounts for these potential biases by relying on an "unbiased" set of anchor items. The PERSUADE corpus does not include such data, however, and there is no public dataset currently available that would permit DIF analyses.

Results showed that demographic affiliations were embedded in the hidden layer of the XLNet scoring model. Yet, without further investigation, we are unable to determine if this information is used (e.g. as an implicit feature) in generating students' essay scores.

Lastly, we recognize that this study was limited to analyzing biases within a single LLM model and dataset. Further research could evaluate other state-of-the-art models and diverse essay sets to determine the extent to which findings generalize.

4.3 Further Research

The limitations of this study, noted above, reveal several promising paths forward. There is room, for instance, to explore additional LLM models (beyond XLNet) and additional datasets. It would also be valuable to investigate sources of group differences (e.g. language differences between groups), and to determine if these group differences are construct relevant or not. Construct (ir)relevance is important to consider, as it affects which debiasing strategies would be viable (Kwako, 2023).

Along the lines of debiasing, it would be helpful to explore bias mitigation techniques at both the training and scoring stages. For example, if demographic affiliation is an implicit feature (i.e. $\mathcal{L}(x) = \alpha x + \beta$, and $\tilde{\mathcal{L}}(x) = \tilde{\alpha} x + \tilde{\beta}$), then we could potentially use orthogonal projection to optimize α on the vector-subspace orthogonal to $\tilde{\alpha}$. This might mitigate the effect of any features the model is using to distinguish demographic information. This may, however, come at some cost to model performance.

References

William H Angoff. 1993. Perspectives on differential item functioning methodology. In *Differential item functioning*, pages 3–23. Routledge.

Robert M Bell and Daniel F McCaffrey. 2002. Bias reduction in standard errors for linear regression with multi-stage samples. *Survey Methodology*, 28(2):169–182.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. ArXiv:2004.05150 [cs].

	Text							Score Features							
#	G	W	L	B	SES	ELL	DS	G	W	L	B	A	SES	ELL	DS
1	0.16	0.25	0.01	0.21				0.07	0.08	0.05	0.14	-0.01			
2	0.22	0.36	0.25	0.07	0.31	0.16	0.13	0.19	0.18	0.09	0.09	0.16	0.17	0.33	0.28
3	0.23	0.26	0.24	0.11	0.30	-0.00	0.00	0.24	0.29	0.16	0.04	0.07	0.18	0.09	0.20
4	-0.08	0.05	0.00	0.22	0.28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.26	0.35	0.24	0.10	0.33	0.42	0.05	0.26	0.26	0.18	0.21	0.14	0.28	0.28	0.23
6	0.17	0.43	0.45	0.10	0.46	0.77	0.02	0.24	0.34	0.38	0.14	0.06	0.28	0.75	0.15
7	0.27	0.18	0.10	0.16	0.17	0.29	0.00	0.22	0.14	0.08	0.18	0.14	0.19	0.21	0.07
8	0.23	0.27	0.24	0.09	0.34	0.30	0.00	0.25	0.25	0.16	0.13	0.05	0.21	0.13	0.13
9	0.26	0.29	0.18	0.12		0.06		0.27	0.24	0.10	0.11	0.07		0.23	
10	0.26	0.25	0.20	0.12	0.32	0.21	0.00	0.19	0.28	0.11	0.13	0.06	0.20	0.15	0.05
11	0.28	0.20	0.06	0.15	0.23	0.00	0.00	0.21	0.14	0.00	0.21	0.00	0.18	0.00	0.00
12	0.32	0.17	0.16	0.00				0.21	0.15	0.08	0.00	-0.05			
13	0.31	0.12	0.08	0.12	0.04	0.00	0.00	0.20	0.05	0.07	0.14	0.13	0.07	0.00	0.00
14	0.28	0.18	0.21	0.08	0.37	0.35	0.01	0.28	0.12	0.08	0.08	0.14	0.26	0.28	0.10
15	0.25	0.21	0.11	0.16	0.19	0.00	0.00	0.06	0.11	0.06	0.09	0.04	0.09	0.14	0.04

Table 4: The ability of our models to determine demographic affiliation measured by Cohen's kappa statistic. The columns under Text present κ values for language models trained on the text, while the columns under Score Features are linear models whose features coincide with those used to determine score.

Yoav Benjamini and Yosef 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. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.2517-6161.1995.tb02031.x.

Rishabh Bhardwaj, Navonil Majumder, and Soujanya Poria. 2021. Investigating Gender Bias in BERT. *Cognitive Computation*, 13(4):1008–1018.

Graeme Blair, Jasper Cooper, Alexander Coppock, Macartan Humphreys, and Luke Sonnet. 2024. *estimatr: Fast Estimators for Design-Based Inference*. R package version 1.0.2.

Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46. Publisher: SAGE Publications Inc.

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 (PERSUADE) corpus 1.0. *Assessing Writing*, 54:100667.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. ArXiv:1901.02860 [cs, stat].

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Technical Report arXiv:1810.04805, arXiv. ArXiv:1810.04805 [cs] type: article.

Daniel E. Ho, Kosuke Imai, Gary King, and Elizabeth A. Stuart. 2011. MatchIt: Nonparametric preprocessing for parametric causal inference. *Journal of Statistical Software*, 42(8):1–28.

Alexander Kwako, Yixin Wan, Jieyu Zhao, Mark Hansen, Kai-Wei Chang, and Li Cai. 2023. Does bert exacerbate gender or l1 biases in automated english speaking assessment? In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 668–681.

Alexander James Kwako. 2023. *Mitigating Gender and L1 Biases in Automated English Speaking Assessment*. Ph.D. thesis, University of California, Los Angeles.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. ArXiv:1711.05101 [cs, math].

Elijah Mayfield and Alan W Black. 2020. Should You Fine-Tune BERT for Automated Essay Scoring? In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 151–162, Seattle, WA, USA → Online. Association for Computational Linguistics.

Mary L. McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia Medica*, 22(3):276–282.

Christopher Ormerod, Amy Burkhardt, Mackenzie Young, and Sue Lottridge. 2023. Argumentation Element Annotation Modeling using XLNet. ArXiv:2311.06239 [cs].

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*.

Christopher M. Ormerod, Akanksha Malhotra, and Amir Jafari. 2021. Automated essay scoring using efficient transformer-based language models. Number: arXiv:2102.13136 arXiv:2102.13136 [cs].

Christopher Michael Ormerod. 2022. Mapping Between Hidden States and Features to Validate Automated Essay Scoring Using DeBERTa Models. *Psychological Test and Assessment Modeling*, 64(4):495–526.

James E. Pustejovsky and Elizabeth Tipton. 2018. Small-Sample Methods for Cluster-Robust Variance Estimation and Hypothesis Testing in Fixed Effects Models. *Journal of Business & Economic Statistics*, 36(4):672–683. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/07350015.2016.1247004.

R Core Team. 2023. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Pedro Uria Rodriguez, Amir Jafari, and Christopher M. Ormerod. 2019. Language models and Automated Essay Scoring. Number: arXiv:1909.09482 arXiv:1909.09482 [cs, stat].

Mark D. Shermis. 2014. State-of-the-art automated essay scoring: Competition, results, and future directions from a United States demonstration. *Assessing Writing*, 20:53–76.

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.

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.

David M. Williamson, Xiaoming Xi, and F. Jay Breyer. 2012. A Framework for Evaluation and Use of Automated Scoring. *Educational Measurement: Issues and Practice*, 31(1):2–13. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1745-3992.2011.00223.x.

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.

Klaus Zechner. 2019. Summary and outlook on automated speech scoring. In *Automated speaking assessment*, pages 192–204. Routledge.

A Differences across student groups

This appendix reports descriptive statistics of essays written by students, disaggregated by demographic affiliations. In addition to known discrepancies between the lengths of essays between certain groups (notably male and female students), we present the average word length, number of sentences, and the Flesch-Kincaid grade, which is a common readability measure defined by

$$\begin{aligned} G \;=\; & \alpha\left(\frac{\text{total words}}{\text{total sentences}}\right) \\ & +\beta\left(\frac{\text{total syllables}}{\text{total words}}+\gamma\right)+\gamma \qquad (3) \end{aligned}$$

where $\alpha = 0.39$, $\beta = 11.8$, and $\gamma = 15.59$. These statistical differences in essay texts, by demographic affiliations, are presented in Table 5.

			Averages				
Category	Subgroup	Rep.	Score	Words	Word Len.	Sent.	F.K.
Gender	Male	49.5%	3.20	404	4.40	19.3	9.32
	Female	50.5%	3.43	432	4.45	21.9	8.70
Race/ Ethnicity	White	44.5%	3.42	427	4.41	21.4	8.60
	Hispanic/Latino	25.2 %	3.08	398	4.40	19.0	9.50
	Black/African American	19.1%	3.12	393	4.43	19.3	9.26
	Asian/Pacific Islander	6.7%	3.37	504	4.59	25.1	9.22
	Two or More	3.9%	3.45	429	4.46	21.1	8.87
	Native American	0.5%	3.02	369	4.35	19.3	8.31
ELL	Identified	8.6 %	2.69	374	4.42	16.5	10.7
	Not Identified	86.4%	3.35	421	4.42	20.9	8.87
Economic Disadvantage	Identified	37.1 %	2.98	367	4.36	18.0	9.19
	Not Identified	42.8%	3.65	446	4.44	22.0	8.9
Disability Status	Identified	10.3%	2.72	360	4.36	17.0	9.6
	Not Identified	69.8%	3.33	416	4.41	20.6	8.95

Table 5: Some key statistical differences between the nature of the scores and essays, disaggregated by demographic affiliation.

Automated Scoring of Clinical Patient Notes: Findings From the Kaggle Competition and Their Translation into Practice

Victoria Yaneva[1], King Yiu Suen[1], Le An Ha[2], Janet Mee[1], Milton Quranda[1], and Polina Harik[1]

[1]National Board of Medical Examiners, Philadelphia, USA
{vyaneva, ksuen-temp,jmee, mquranda, pharik} @nbme.org
[2]Ho Chi Minh City University of Foreign Languages, Vietnam
anhl@huflit.edu.vn

Abstract

Scoring clinical patient notes (PNs) written by medical students is a necessary but resource-intensive task in medical education. This paper describes the organization and key lessons from a Kaggle competition on automated scoring of such notes. 1,471 teams took part in the competition and developed an extensive, publicly available code repository of varying solutions evaluated over the first public dataset for this task. The most successful approaches from this community effort are described and utilized in the development of a PN scoring system. We discuss the choice of models and system architecture with a view to operational use and scalability, and evaluate its performance on both the public Kaggle data (10 clinical cases, 43,985 PNs) and an extended internal dataset (178 clinical cases, 6,940 PNs). The results show that the system significantly outperforms a state-of-the-art existing tool for PN scoring and that task-adaptive pretraining using masked language modeling can be an effective approach even for small training samples.

1 Introduction

A core practice in assessing the clinical skills of medical students is the use of Objective Structured Clinical Examinations (OSCEs) – a type of exam, where test-takers interact with *standardized patients*, who are trained to portray a set of clinical symptoms. After examining the patients, the test-takers are asked to describe their findings in a clinical patient note (PN), similar to those found in electronic health records (see an example PN in Appendix A). The PN serves as a documentation of the encounter and is used to assess examinee ability to gather information, record physical examinations, and interpret clinical data. OSCEs are widely used in medical schools in various countries, with around 90% of US schools requiring their students to pass such exams (Barzansky and Etzel, 2016).

A major bottleneck for scaling OSCE assessment is the time, cost, and effort associated with the expert grading of large amounts of PNs, especially given limited faculty time. For example, in the former United States Medical Licensing Examination® (USMLE®) Step 2 Clinical Skills exam (discontinued in 2020), more than 100 licensed physician raters were needed every year to grade ≈ 330,000 PNs from ≈ 35,000 US and international test-takers (Sarker et al., 2019).

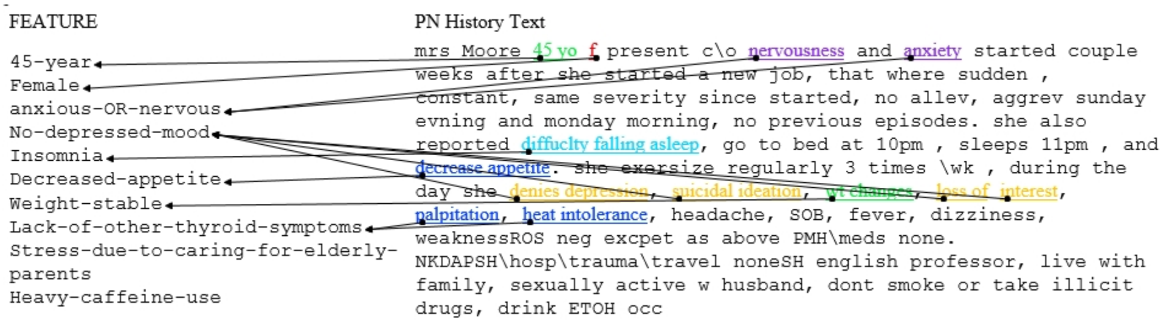
While there is interest among medical educators to address the above limitations using automated grading methods, the exploration of such methods has been slow and fragmented due to exam security concerns, which limit data sharing. This has resulted in small-scale, predominantly internal explorations of automated scoring, with no shared datasets or code to foster collaborative research.

To address this gap, we organized a Kaggle competition on clinical PN scoring[1] as a community effort to move this field forward. We then used the most successful approaches for the development of an interpretable and transparent PN scoring system. The contributions of this paper are as follows:

- Description of the Kaggle competition on clinical PN scoring, for which we released a public dataset and which resulted in a large repository of publicly available code.
- Analysis of the most successful approaches.
- Description of an Amazon Web Services (AWS) proof-of-concept for PN scoring based on the successful solutions; Choice of models and system architecture are discussed with a view to operational scalability. Models are

[1] https://www.kaggle.com/c/nbme-score-clinical-patient-notes/overview

June 20, 2024 ©2024 Association for Computational Linguistics



mrs Moore 45 yo f present c\o nervousness and anxiety started couple weeks after she started a new job, that where sudden , constant, same severity since started, no allev, aggrev sunday evning and monday morning, no previous episodes. she also reported diffuclty falling asleep, go to bed at 10pm , sleeps 11pm , and decrease appetite. she exersize regularly 3 times \wk , during the day she denies depression, suicidal ideation, wt changes, loss of interest, palpitation, heat intolerance, headache, SOB, fever, dizziness, weaknessROS neg excpet as above PMH\meds none. NKDAPSH\hosp\trauma\travel noneSH english professor, live with family, sexually active w husband, dont smoke or take illicit drugs, drink ETOH occ

Figure 1: Example of rubric features and their annotated expressions within a patient note excerpt

trained in a real-world scenario of limited PNs for newly developed cases.

- Baseline comparison to an already operationalized scoring system with a mean F1 score improvement from .76 to .95. Performance is evaluated on both the Kaggle data and an extended internal dataset.
- Evaluation of a scenario when training is performed using limited annotation.
- Discussion of ethical considerations and implications for fairness, reliability, and validity.

2 Context

The data used in this study originated from the United States Medical Licensing Examination® (USMLE®) – a series of examinations used to support medical licensure decisions in the United States. Until 2020, the USMLE Step 2 Clinical Skills examination was a part of the USMLE step sequence and involved interactions with standardized patients portraying different clinical scenarios. The resulting PNs were graded using rubrics specific to each clinical case, which contain a set of *features* – important concepts, which should appear in an appropriately documented PN (Figure 1). For example, for a clinical case about a patient with anxiety, it may be important that the examinee discovers that the patient has *insomnia*, in which case *insomnia* would be listed as a rubric feature. PNs that do not mention that symptom or some expression of it such as *difficulty falling asleep* would receive a lower rater score.

Key challenge for automated scoring is the variety of ways features are expressed (e.g., *evaluation for coronary risk factors* expressed as *father with MI at age 50*, or *denies depressed mood* expressed as *(-) anhedonia*). There are cases of ambiguous negation as in *denies nausea, vomiting* for the feature *no nausea and or vomiting* or temporal aspects such as *recent URI* for *uri one week ago*. To be operationally usable, a PN scoring system needs to provide interpretable evidence and be highly accurate. These requirements are crucial to ensure exam fairness and protect the health of the public.

3 Related Work

The vast majority of work on automated scoring has been done in the field of writing evaluation (see Klebanov and Madnani (2020) for an overview). Studies on scoring clinical text include Latifi et al. (2016), who use a feature-based system for scoring short responses to clinical decision-making questions, Ha et al. (2020) who predict examinee proficiency from responses to clinical short-answer questions, and Suen et al. (2023) who use transformer models for scoring short answers to clinical questions. For PN scoring specifically, Yim et al. (2019) use features and BERT embeddings for scoring a corpus of 338 PNs and Zhou et al. (2022) use weakly supervised approaches and transfer learning for scoring two clinical cases of 30 PNs each.

The work most relevant to ours is the INCITE system (Sarker et al., 2019; Harik et al., 2023), which was developed for operational scoring of PNs from the USMLE Step 2 CS exam and which we use as a baseline. The system is a modular pipeline which outputs a binary score of "found" or "not found" for each rubric feature, utilizing custom-built lexicons and annotations. The first two modules perform direct and fuzzy matching between a feature or a lexicon variant and the PN text using a fixed or dynamic Levenshtein ratio threshold. Any features whose expressions are found using this method are removed from the pipeline to optimize running time. Next, matching is performed against combinations of lexicon variants and annotations, which "often leads to an explosion of the number of eventual entries" (Sarker et al., 2019) as terms in the annotations are replaced with variants from the lexicons. To limit this search space, there is a cap of 10,000 randomly sampled combinations per feature. Matching is then done using these phrases as sequences and as bag-of-words to cope with fragmented entries [2].

Advantages of the INCITE system include its high performance, ability to be tuned for precision and recall by varying the thresholds, as well as its speed – it is capable of processing over 50,000 PNs per day on a desktop computer. However, the rule-based nature of the system limits improvement from more training data, especially because more annotations would greatly increase the search space for supervised concept detection.

4 Task description and evaluation

The task of developing an interpretable system for automated scoring of PNs is one where features from the rubric are mapped to expressions from the PN. If an expression of the feature is identified in the PN then the feature is considered "found", else it is "not found". The more features are found, the higher the score for that PN. We perform two types of model evaluation, as described below.

Token-level evaluation: This type of evaluation answers the question "What phrase spans in the PN correspond to a given rubric feature?". This evaluation is identical to the one used in the Kaggle competition and comparable to its leaderboard.

For each instance, the system predicts a set of character spans that it considers to correspond to that feature, where a character span is a pair of indexes representing a range of characters within a text. These predicted spans are then compared to ground-truth spans from the annotation and scored as: a character is considered true positive if it is within both a ground-truth and a prediction; false negative if it is within a ground-truth but not a prediction; and false positive if it is within a prediction but not a ground truth. An overall F1 score is computed from the TPs, FNs, and FPs aggregated across all instances[3].

Binary evaluation: This type of evaluation answers the question "Was an expression of a feature found (1) or not found (0) in the PN?". This evaluation corresponds to the way PNs are scored in practice.[4] If at least one span is identified as corresponding to the feature, the feature is considered "found". For the neural models, binary scores are obtained by applying a function over the token-level predictions using a threshold of 0.5.

5 Data

Training and evaluation are performed in two datasets of PN history portions[5] – public and proprietary – from the USMLE Step 2 CS exam.

Public dataset: This dataset was used in the Kaggle competition (so henceforth referred to as "the Kaggle dataset") and contains the history portions of 43,985 PNs from 10 clinical cases and the corresponding features for each case. Data were collected between 2017 and 2020 from 35,156 US or international test-takers who took the exam under standardized conditions in one of five testing locations in the US. The average number of PNs per case is 4,398 (min = 992, max = 9,936), total number of tokens is 5,958,464, and the average length of each history portion is 135.47 tokens (SD = 24.27). The average number of features per history portion is 14.3 (SD = 3.34). Of these, a total of 2,840 PNs (284 per case) were annotated by 10 experienced US medical practitioners who were asked to identify the spans of each phrase that is an expression of a rubric feature and link it to that feature. The annotators were divided in pairs of two and 20% of the PNs from each case were double-rated (see detailed annotation guidelines and procedure in Appendix B). F1 agreement scores were computed using the token-level evaluation procedure described above and showed a substantial agreement across all cases (F1 = .84 (SD = 0.075); Cohen's κ of 0.89 (SD = 0.057)). Binary F1 denoting whether an expression of a given feature was found in a PN was F1 = 0.97 (SD = 0.014). Detailed information about the corpus can be found in Yaneva et al. (2022). The data is available via a data sharing agreement at https://www.nbme.org/services/data-sharing.

Proprietary dataset: This dataset consists of a much larger number of clinical cases – 178 – with

[2] E.g., "Antibiotics taken in recent times for his symptoms – negative". As Sarker et al. (2019) note, window-based fuzzy matching would fail to include the negation and the rest of the description in one window.

[3] For specific examples, see https://www.kaggle.com/competitions/nbme-score-clinical-patient-notes/overview/evaluation

[4] Raters are not typically required to mark the exact expressions that correspond to a feature. As a result, human scores are not explicitly traceable to specific evidence in the PN, unless this is specifically required (e.g., if a score is contested).

[5] The history portion is where all relevant clinical information obtained from an interview with the patient is described.

fewer PN history portions per case (μ = 39; min = 32; max = 43). Total number of PNs in the set is 6,940, and total number of tokens is 1,121,236. Average document length is 161.56 tokens (SD = 29.42), and the average number of features is 13.92 (SD = 4.84). All PNs were annotated following the same procedure as above, resulting in binary inter-annotator agreement of F1 = .95 (SD = 0.09), computed over 10% double-rated notes per case.

6 Kaggle competition: Key lessons

The Kaggle competition on scoring clinical PNs resulted in a total of 28,049 code entries from 1,471 participating teams. After the end of the competition, many teams posted their notebooks in the competition's code repository, making them publicly available. In this section, we analyze the top 15 publicly shared solutions [6] (the teams ranking from 1st to 11th place, and those that ranked #13, #14, #18, #19, and #20), as well as insights from other notebooks and key forum discussions. The final leaderboard rankings can be seen at `https://tinyurl.com/p9mwfu8c` and corresponding code contributions can be accessed at `https://tinyurl.com/3h8p5a67`.

Results Many of the top-performing teams reached a token-level F1 score of .89, with minor differences between solutions (e.g., #1 F1 = .89456, #2 F1 = .89432, and #3 F1 = .89384), indicating that there are different, equally successful ways of addressing this task. This result also suggests potential ceiling effects arising from annotation inconsistencies such as not capturing every instance of a phrase that can be mapped to a feature[7] or not identifying the correct character span of a phrase (average inter-annotator agreement F1 = .84). Such inconsistencies resulting from human error are inevitable in spite of rigorous training and data cleaning efforts, further showcasing the need for improved reliability in scoring.

Key approaches Most high-performing solutions used some version of DeBERTa (He et al., 2021) as the backbone and performed **task-adaptive pretraining** (Gururangan et al., 2020) by using masked language modeling (MLM) over the unannotated portion of the data. One solution (#2) additionally pretrained on the SQuAD 2.0 question answering dataset (Rajpurkar et al., 2018), drawing a parallel between the two tasks: the feature text in PN scoring corresponds to the *question* in the SQuAD data, the patient history is the *context*, and the annotations are all *answers* to the question.

Another approach shared by almost all of the analyzed solutions was the use of **pseudo labeling** (Arazo et al., 2020) to create more training data from the unannotated notes. One team (ranking #3) also utilized **meta pseudo labeling** (Pham et al., 2021). Some teams reported that hard labels work better than soft labels [solutions ranking #8, #70], while others reported the opposite [#1].

While these techniques were used in most high-performing solutions, one approach that distinguished the Top 3 winners was the use use of **multi-task learning**. In this case, the main task of token classification is combined with an auxiliary task of predicting annotation span boundaries, putting more weight on tokens that are the beginning or end of a phrase. In the model architecture, this is expressed as a primary head for token classification and two auxiliary heads for span boundary detection (one for starts and one for ends).

A focal point for most successful solutions was the prevention of overfitting. This was done through careful ensembling and detailed experimentation with various dropout rates, as well as extensive use of cross validation.

7 Models

Two key differences between real-world applications and the competition are that: i) newly developed cases do not come with large amounts of unannotated PNs (which makes pseudo-labeling not suitable), and ii) the trade-off between performance gain and resource requirements such as speed and compute power is an important aspect of model selection (making the ensembling of a large number of models impractical). With these prerequisites in mind, the following approaches were trained and evaluated.

INCITE baseline: The INCITE system is an operationally used benchmark. The case-specific data in its lexicons is from the training set for each case.

DeBERTa baseline: The pretrained DeBERTa v3 (He et al., 2021) was used as the backbone

[6] Detailed solution descriptions for first place: `https://tinyurl.com/2p8afa94`, second place: `https://tinyurl.com/yc77s4rk`, and third place `https://tinyurl.com/3yf4u6hr`.

[7] The 2nd place winner hypothesised that annotators were more likely to miss repeated annotations than first occurrence and noted that the use of recursive neural networks (RNNs) could be useful to capture such sequence dependencies.

Token-level results for the public (Kaggle) dataset

	Public test set			Private test set		
	P	R	F1	P	R	F1
DB	**.846**	.882	.864	.85	.885	.867
DB + MTL	.844	.882	.862	**.849**	.887	.868
DB + MLM	.845	**.889**	**.866**	**.849**	**.89**	**.869**

Token-level results for the proprietary data (178 cases)

	Training set (80%)			Test set (20%)		
	P	R	F1	P	R	F1
DB	.836	.782	.808	.681	.768	.722
DB + MTL	.845	.808	.826	.773	.7	.745
DB + MLM	**.94**	**.95**	**.945**	**.856**	**.834**	**.845**

Table 1: Token-level results. DB = DeBERTa; MTL = multi-task learning; MLM = masked language modeling; P = precision, R = recall. Note that INCITE does not output token-level information.

model[8], where each token was assigned a label of 1 if inside the annotation span and 0 otherwise. The output of the model was the probability of each token being inside the annotation span. After experimentation with various probability thresholds in both datasets, a threshold of 0.5 was determined sufficient (i.e., a token with a probability greater than 0.5 was considered to be inside the span). The model was trained with cross-entropy loss.

DeBERTa + Masked Language Modeling (MLM): 15% of the tokens in the input sentences were randomly masked and ran through the model, where the model's objective was to predict the masked tokens. For the Kaggle dataset, the pretraining was performed on the unlabeled data. For the proprietary dataset, there were no unlabeled data, so the pretraining was performed on the labeled data from the training set. The MLM model was pretrained for one epoch. The pretrained model was then trained the same way as the baseline model.

DeBERTa + Multi-task Learning (MTL): Two auxiliary tasks were trained jointly with the model, predicting whether the token was at the beginning (Task 1) or the end (Task 2) of the annotation span.

8 Results

Token-level results: Table 1 presents the results from the token-level evaluation. For the Kaggle data, we kept the exact training, private test, and public test sets,[9] so the results are directly comparable to the competition leaderboard. As shown in the table, the best-performing model is DeBERTa + MLM, with a private test set F1 score of .869 (P = .849, R = .89). This compares to F1 = .89456 for the #1 Kaggle solution. A drop in performance of only .03 points shows that the exclusion of pseudo-labeling and the use of a single model instead of an ensemble of multiple models did not lead to a loss that has a practical significance (although such difference is important in a competition context).

For the internal dataset the results are consistent with Kaggle – the best model is again DeBERTa + MLM (F1 = .845, P = .856, R = .834). The model generalizes over a much larger set of cases and is robust when trained on fewer notes (as a reminder, the internal dataset contains 32 to 49 annotated notes per case (80% used for training), compared to 100 training notes per case in Kaggle). Importantly, this result shows that MLM pretraining can be fruitfully applied to small training sets, leading to an increase over the DeBERTa baseline (.845 vs. .722). The DeBERTa and DeBERTa + MTL results did not generalize as well, exemplifying the importance of task-adaptive pretraining.

Note that token-level evaluation was only performed with the neural models. INCITE cannot output specific phrases if the matching was done by some of its more advanced modules (e.g., bag of words from lexicon variants + fuzzy matching). This is an important distinction between INCITE and the neural approaches that has implications for both interpretability and intended use (e.g., in providing feedback to learners).

Binary evaluation results and comparison to INCITE: The binary evaluation results are presented in Table 2. For Kaggle, the neural models outperform INCITE (F1 of .958 for DeBERTa + MLM; .888 for INCITE on the public test set). This difference is more pronounced for the proprietary dataset, where DeBERTa + MLM's robust F1 of .952 compares to an F1 of .761 for INCITE and .946 for inter-annotator agreement. As shown, the main difference with INCITE is that DeBERTa + MLM has a much higher recall (e.g., R = .954 vs. R = .642 for INCITE). Precision is high for both DeBERTa + MLM (.95) and INCITE (.953).

The binary evaluation results on the internal

[8] Learning rate: 1e-4; Optimizer: AdamW; Weight decay: 0.01; Learning rate scheduler: Linear (warmup for 10% of the training steps); Training epochs: 5; Training batch size: 4 per device x 2 GPUs = 8; Gradient accumulation steps: 4.

[9] In Kaggle, the public test set serves as a validation set for the development of the approaches. The final leaderboard is determined after the end of the competition by the performance of the submitted models on the private test set.

Binary evaluation for the public (Kaggle) dataset

	Public test set			Private test set		
	P	**R**	**F1**	**P**	**R**	**F1**
INCITE	**.962**	.818	.883	**.961**	.828	.888
DB	.95	**.962**	.956	.951	**.963**	.957
DB + MTL	.947	.961	.954	.953	**.963**	**.958**
DB + MLM	.952	.961	**.957**	**.961**	.956	**.958**

Binary evaluation for the proprietary data (178 cases)

	Training set (80%)			Test set (20%)		
	P	**R**	**F1**	**P**	**R**	**F1**
INCITE	.966	0.85	.902	**.953**	.642	.761
DB	.927	.933	.93	.896	.862	.879
DB + MTL	.933	.947	.94	.898	.877	.888
DB + MLM	**.979**	**.979**	**.979**	.95	**.954**	**.952**

Table 2: Binary evaluation results. DB = DeBERTa; MTL = multi-task learning; MLM = masked language modeling, P =precision, R = recall.

dataset for DeBERTa + MLM (F1 = .952) are comparable to human-rater performance as computed on the set of double-rated PNs per case (inter-rater agreement F1 = .946).

Out of the total of 19,465 instances, INCITE and DeBERTa + MLM agreed in 14,532 or 75% of the instances (κ = 0.51, indicating moderate agreement); INCITE vs Annotation agreement was κ = 0.52; Finally, DeBERTa + MLM vs Annotation agreement was κ = 0.89.

Limited annotation setting: For the internal dataset we also experiment with a limited annotation setting, since the question of how much annotation is required before a model can be trained has strong practical implications. For a limited annotation scenario where we train on 30% of the data (i.e., $\approx$ 12 PNs per case) and evaluate on 70% held-out data, the F1 score for DeBERTa + MLM is .836 (binary F1 = .94) compared to .69 (binary F1 = .83) for DeBERTa + MTL and .64 (binary F1 = .86) for DeBERTa baseline. These results show that task-adaptive pretraining leads to robust models even in a limited annotation scenario.

9 Error Analysis

For DeBERTa + MLM, there were 990 errors (594 FNs and 396 FPs), distributed across all 178 clinical cases[10]. However, the errors were only distributed across 36% of the 1815 features. We hypothesize that this may be due to differences in annotation length for different features. Indeed, the mean annotation length differs between the correct predictions and the errors: it is μ = 19.6 (SD = 20.7) for correct and μ = 13.2 (SD = 17.2) for the errors (Mann-Whitney U = 7183226, p < 0.001). This is somewhat counter-intuitive, as it suggests that the shorter features and shorter annotations are more difficult to detect. Further content-specific analysis is needed to illuminate the potential causes for this phenomenon. Annotation length affected INCITE inversely and to a much greater extent, where the annotations for the correct class (μ = 15.9, SD = 19.8) are on average twice as short as the errors (μ = 29.12, SD = 19.9), (U = 19231079.5, p = 0.0), potentially due to limitations from its window-based approach. Spearman correlation between annotation length and correct/incorrect predictions further supports this finding: r = 0.08 for DeBERTa + MLM model and r = -0.36 for INCITE. A likely explanation for this result is that INCITE's window-based approach is challenged by long phrases, while DeBERTa's multi-head self-attention layers, where the encoder reads the entire sequence bidirectionally, enables it to cope well with these. In addition, since the objective of the neural models was to decide whether a given character belongs to a relevant phrase, the higher character count of longer phrases increases the available information for making a prediction. Further analysis of the differences between correct and erroneous predictions did not reveal a specific pattern. This extended analysis is presented in Appendix D together with examples of specific features.

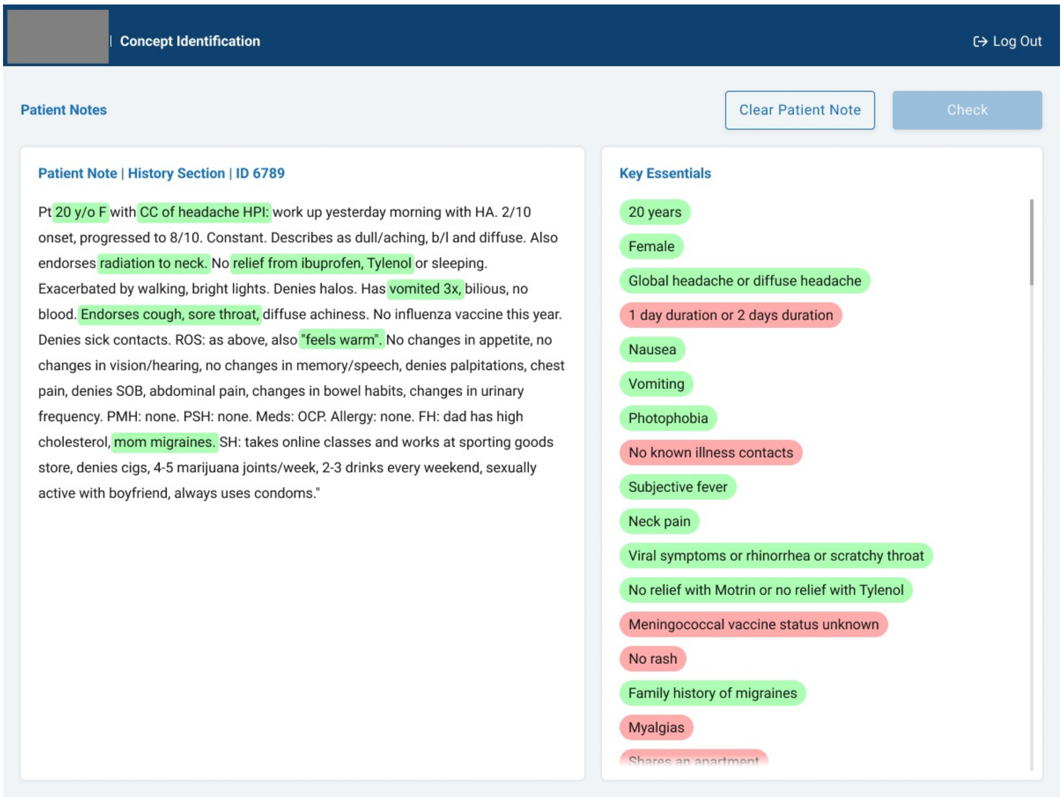
10 Deployment

A system based on the DeBERTa + MLM model was deployed on the Amazon Web Services (AWS) platform. A graph depicting the AWS architecture can be seen in Appendix C. Figure 2 provides a visualization of the system output. Speed, resource efficiency, and scalability are ensured by the use of SageMaker and eliminating the need for human interference via event triggers: placing incoming data in an initial S3 bucket triggers a series of Lambda functions, which initiate preprocessing, training, and scoring.

11 Discussion

The results presented above showed that the best model, DeBERTa + MLM, led to significant im-

[10] The average number of errors per case was 10.5 (SD = 9.15), with 4 cases scored without any errors, 16 cases with one, and 22 cases with two errors; highest number of errors in a case was 18 (1 case), followed by 17 (1 case), and 15 (3 cases). The number of errors per case (μ = 10.5 (SD = 9.15)) was best explained by the number of features in a case, where cases with higher number of features had more errors.



Pt 20 y/o F with CC of headache HPI: work up yesterday morning with HA. 2/10 onset, progressed to 8/10. Constant. Describes as dull/aching, b/l and diffuse. Also endorses radiation to neck. No relief from ibuprofen, Tylenol or sleeping. Exacerbated by walking, bright lights. Denies halos. Has vomited 3x, bilious, no blood. Endorses cough, sore throat, diffuse achiness. No influenza vaccine this year. Denies sick contacts. ROS: as above, also "feels warm". No changes in appetite, no changes in vision/hearing, no changes in memory/speech, denies palpitations, chest pain, denies SOB, abdominal pain, changes in bowel habits, changes in urinary frequency. PMH: none. PSH: none. Meds: OCP. Allergy: none. FH: dad has high cholesterol, mom migraines. SH: takes online classes and works at sporting goods store, denies cigs, 4-5 marijuana joints/week, 2-3 drinks every weekend, sexually active with boyfriend, always uses condoms."

- 🟩 20 years
- 🟩 Female
- 🟩 Global headache or diffuse headache
- 🟥 1 day duration or 2 days duration
- 🟩 Nausea
- 🟩 Vomiting
- 🟩 Photophobia
- 🟥 No known illness contacts
- 🟩 Subjective fever
- 🟩 Neck pain
- 🟩 Viral symptoms or rhinorrhea or scratchy throat
- 🟩 No relief with Motrin or no relief with Tylenol
- 🟥 Meningococcal vaccine status unknown
- 🟥 No rash
- 🟩 Family history of migraines
- 🟥 Myalgias
- 🟥 Shares an apartment

Figure 2: System output for an example PN

provements over INCITE for a diverse set of 178 clinical cases (binary F1 = .95 for DeBERTa + MLM compared to .76 for INCITE), as well as the Kaggle data (.96 vs .89). INCITE was significantly more challenged by lengthy phrases and the smaller number of training instances in the proprietary dataset. By contrast, as shown when evaluating in the limited annotation scenario, DeBERTa + MLM continues to yield meaningful gains when trained on as few as 12 PNs. These experiments add evidence that task-adaptive pretraining can be beneficial even for small training samples, making the approach applicable to a wide range of practical scenarios.

While the INCITE system struggled to identify lengthy expressions (i.e., the annotations of the errors were twice as long as those of the correctly identified instances), the DeBERTa + MLM model coped well with long sequences. This is likely due to the multi-head self attention layers of DeBERTa, where the encoder reads the entire sequence in a bidirectional manner. In addition, since the task was to decide whether a given character belongs to a relevant phrase or not, the higher character count of longer phrases increases the available information for making a prediction. At the same time, INCITE's window-based approach limits the length of the text spans being considered at a time, making the capturing of long dependencies less feasible.

The ability of the neural approaches to output the relevant PN phrases that correspond to each feature greatly improves the interpretability of the scoring process by making explicit the relationship between the assigned score and its supporting evidence. Importantly, this is an improvement not only upon INCITE but also upon human scoring, as raters rarely have the time capacity to mark specific expressions. As each human rater scores hundreds of patient notes, it is not practically feasible for them to link specific phrases to rubric features for a large volume of data. In addition to improving interpretability, outputting the phrases enables applications of these tools that go beyond summative assessment. Such information can serve to provide pointed learner feedback in OSCE assessment, especially in cases where students are still learning how to document their clinical findings in an appropriately detailed and organized manner.

When discussing the development of this system, it is important to mention community competitions as an important source of innovation. The benefits from sharing data for such purposes are not limited to the organization or the data science community, but extend to improving transparency – a crucial prerequisite for building stakeholder trust. When applying these creative approaches to a real-world scenario, important considerations such as speed and scalability limit the use of large model ensembles that are typically widely used in competitions. Other practical considerations include data availability for training (e.g., newly developed cases rarely have large numbers of PNs associated with them) and the need for weak supervision.

12 Limitations and ethical considerations

Some of the limitations of this research relate to the small within-case sample size of the annotated notes (which is somewhat mitigated by the large number of clinical cases) and the fact that not all notes could be double-rated due to resource constraints. While the scoring method is interpretable in that it can be traced to specific phrases within the PNs, the neural algorithms that identify the phrase boundaries are black-box models which needs to be carefully scrutinized for bias. In addition, it is still not fully apparent why certain features are easier to detect than others. Future work includes development of scoring approaches for other segments from the PNs such as the Physical Examination and Data Interpretation sections, deeper exploration of challenges related to specific features, experimentation with adversarial training, as well as further investigation of the operational use of the system.

Like many other products, automated scoring tools are socio-technical systems, whose impact is determined not solely by their technical capabilities but also by their use and output interpretation. Misuse and incorrect interpretation of the model outputs can lead to unethical practices of serious consequence. In a summative setting, the models described here are intended to be used as hybrid systems, where borderline cases and the notes from examinees below the passing standard are always reviewed by human raters. In a formative setting, it is paramount to carefully examine the relationship between use of the system and learning outcomes as necessary validity evidence.

Another ethical consideration for this study is the transparency of the approaches when developing technology for highly consequential decisions. As Spadafore and Monrad (2019) write: “decisioning software with the potential to profoundly affect the career of a medical student should be examined closely. Transparency of implementation is critical for such a high-stakes application". This is particularly important in automated scoring, where the scores only have value if all stakeholders (e.g., faculty, students, and residency selection programs, to name a few) trust that they are fair, reliable, and valid. Having public datasets and code such as the ones shared in the Kaggle competition go a long way in building trust by increasing transparency and accountability. As per the rules of the Kaggle competition[11], all code shared publicly is licensed under an Open Source Initiative-approved license. It is important to note that the benefits of system transparency go hand-in-hand with risks associated with using that knowledge to "game" the system. These include reverse-engineering a strategy that would result in a higher score, as well as the occurrence of negative “washback" (Green, 2013) – over-focus on developing only those skills that are currently covered by the scoring tool. Limiting these negative consequences while also building trust through transparency requires a delicate balance. In the case of this study, we foster transparency via organizing the competition, describing the main approaches, and evaluating our system on a dataset we made public. At the same time, we do not publish the code behind the system, limiting potential efforts reverse-engineer it or "game" it.

The data used in the Kaggle competition was released following strict adherence to ethical practice. It contains PNs only from examinees who explicitly indicated that they agreed to have their data used in research as part of the official exam registration process; Use of the anonymized data was considered “exempt" following an IRB review. The PNs were assigned a new set of IDs that cannot be linked to operational IDs used in scoring. None of the PNs include names, affiliations or personal descriptions (note that the names and clinical data associated with the standardized patients do not belong to real people; they are part of carefully constructed clinical cases that aim to resemble real-world clinical practice). In addition, the dataset does not feature complete PNs (only history portions are included), and no identifying information is given on which PNs were written by an individual examinee. According to Kaggle’s terms and conditions, data can only be accessed for partici-

[11] https://www.kaggle.com/competitions/nbme-score-clinical-patient-notes/rules

pating in the competition. For purposes unrelated to the competition, access to the data is subject to an application process and a data use agreement as a way to ensure ethical use.

A few important aspects remain to be examined before the system can be used in practice. This includes analyses related to differential functioning of the system for users with different backgrounds, e.g., ensuring that non-native English speakers are not disproportionally penalized due to differences in language proficiency, as well as continuous monitoring for issues such as drift or latency.

References

Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. 2020. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Barbara Barzansky and Sylvia I Etzel. 2016. Medical schools in the united states, 2015-2016. *JAMA*, 316(21):2283–2290.

Anthony Green. 2013. Washback in language assessment. *International Journal of English Studies*, 13(2):39–51.

Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *CoRR*, abs/2004.10964.

Le Ha, Victoria Yaneva, Polina Harik, Ravi Pandian, Amy Morales, and Brian Clauser. 2020. Automated prediction of examinee proficiency from short-answer questions.

Polina Harik, Janet Mee, Christopher Runyon, and Brian E Clauser. 2023. Assessment of clinical skills: a case study in constructing an nlp-based scoring system for patient notes. In *Advancing Natural Language Processing in Educational Assessment*, pages 58–73. Routledge.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

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.

Syed Latifi, Mark J Gierl, André-Philippe Boulais, and André F De Champlain. 2016. Using automated scoring to evaluate written responses in english and french on a high-stakes clinical competency examination. *Evaluation & the health professions*, 39(1):100–113.

Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. 2021. Meta pseudo labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11557–11568.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

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.

Maxwell Spadafore and Seetha U Monrad. 2019. Algorithmic bias and computer-assisted scoring of patient notes in the usmle step 2 clinical skills exam. *Academic Medicine*, 94(7):926.

King Yiu Suen, Victoria Yaneva, Janet Mee, Yiyun Zhou, Polina Harik, et al. 2023. Acta: Short-answer grading in high-stakes medical exams. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 443–447.

Victoria Yaneva, Janet Mee, Le An Ha, Polina Harik, Michael Jodoin, and Alex Mechaber. 2022. The usmle® step 2 clinical skills patient note corpus. Association for Computational Linguistics.

Wen-wai Yim, Ashley Mills, Harold Chun, Teresa Hashiguchi, Justin Yew, and Bryan Lu. 2019. Automatic rubric-based content grading for clinical notes. In *Proceedings of the Tenth International Workshop on Health Text Mining and Information Analysis (LOUHI 2019)*, pages 126–135.

Jianing Zhou, Vyom Nayan Thakkar, Rachel Yudkowsky, Suma Bhat, and William F Bond. 2022. Automatic patient note assessment without strong supervision. In *Proceedings of the 13th International Workshop on Health Text Mining and Information Analysis (LOUHI)*, pages 116–126.

A Example of a patient note

See Table 3 below.

B Annotation Guidelines

For each case, two of the notes were annotated jointly by a pair of annotators as part of an initial discussion to resolve discrepancies in the approach, with the next 5 notes annotated independently and discussed in a follow-up meeting. After that each annotator would proceed to independent work, where a subset of the notes were double-rated

History: Describe the history you just obtained from this patient. Include only information (pertinent positives and negatives) relevant to this patient's problem(s). Karin Moore is a 45 yo F here for nervousness. She recently noticed that she was feeling more nervous than usual and that this feeling has been progressively worsening. Nothing seems to help her nervousness. It is exacerbated by family and work. She feels especially nervous on Sunday night and Monday morning when as she is planning for the week. Unable to fall asleep and doesn't have appetite, though she does makes sure to eat. She denies significant changes in appetite, weight loss, or overall wellbeing. No fevers, chills, nausea, constipation, diarrhea, skin changes, racing heart, shortness of breath, dizziness, headaches or rashes. ROS: otherwise negative PMH: None; PSH: None Meds: Tylenol for occasional HA FHX: Father died at 65yo, had an MI Allergies: NKDA SH: Lives at home with husband, mother, and youngest son. Teaches literature at a local college. Has 2 drinks/mo, no tobacco or drug use.
Physical Examination: Describe any positive and negative findings relevant to this patient's problem(s). Be careful to include only those parts of examination you performed in this encounter. VS: Blood Pressure: 130/85 mm Hg Heart Rate: 96/min Gen: No acute distress, conversational, thin Neck: No thyromegaly, no lymphadeopathy Heart: RRR, no murmurs, rubs or gallops. Radial pulses +2 bilaterally Lungs: Clear to ascultation bilaterally, no wheezes Psych: Well-groomed. Non-pressured speech, linear though process.
Data Interpretation: Based on what you have learned from the history and physical examination, list up to 3 diagnoses that might explain this patient's complaint(s). (...) General anxiety disorder Panic disorder Hyperthyroidism

Table 3: Illustration of a PN. The dataset features only the history portions of the PNs.

for measuring agreement (10% for the proprietary data and 20% for the public data).

The annotators were given the following instruction:

- Identify all phrases that are expressions of a feature from the History portion of the PNs and link them to their corresponding feature.

- Include fragmented annotations by excluding the text that is not relevant to the feature (e.g., if the feature is *No relief with Imodium or Cipro*, only the underlined text of the following excerpt should be annotated: *Has tried Immodium (aggrevated condition), and Cipro 250mg BID (has taken 9 tablets) from prior episode of diarrhea in Kenya of lesser severity (no effect)*)

- Each feature should be marked up as a separate annotation, and the annotation should include all, but not more than, the text that captures the meaning of the corresponding entry in the feature (e.g., if the key essential is *No blood in stool*, only the underlined text of the following excerpt should be annotated: *No blood or mucus in stool*).

- Annotations should include quantifiers (e.g., *twice, four times, some*), intensifiers (e.g., *mild, severe*), and temporal modifiers (e.g., *two weeks, several years*) that are specified in the corresponding entry in the feature, as well as the object that is being described (e.g., *pain, cough*).

- Annotations should not include articles (e.g., *a, the*) or references to the patient (e.g., *her, he*) that occur at the beginning of note entries, or end punctuation (e.g., periods); however, it is not necessary to fragment annotations if words or characters, such as these, occur within relevant text and do not modify the meaning of the feature entry.

- Annotations may overlap; that is, they may share text with other annotations. For example, negations (e.g., *negative for, no, denies*) frequently will be shared among several annotations. In the phrase *Negative for fever, chills, nausea, vomiting, hematochezia*, the negated nouns refer to different features and should be annotated as Negative for fever, *Negative for chills, Negative for nausea*, etc.

- Mark up every instance of the feature whether it is identical to an existing annotation or not. For example, if the feature is *NSAID-use* and the examinee wrote *Uses NSAIDs* as well as *took ibuprofen*, both snippets of text should be annotated. If the exact snippet *Uses NSAIDs* appeared more than once in a note, it should be annotated every time it appears in the note.
- Gender is a special case of a feature and should only be annotated once for the first mention. Subsequent phrases that may be linked to gender such as *she* or *his* should not be annotated.

C AWS System Architecture

See Figure 3 below for a visualization of the system architecture.

D Extended Error Analysis

Examples of features that were always correctly identified include *'no previous uti', 'occasional morning headaches', 'no temperature intolerance or no weight change or no bowel changes or no hair changes or no skin changes', 'on depo provera',* and *'decreased energy or fatigue'*. The top 5 features with most FPs were *getting worse* (7), *hand stiffness* (5), *subjective fever* (5), *chest pain with cough* (5), and *overdue for colonoscopy* (5). The top 5 features that were most difficult to detect automatically with highest numbers of FNs were *1 day urinary frequency* (4), *radiating down back of neck* (3), *constipation x 4 5 months* (3), *acute onset* (3), *nausea* (3). There was no apparent pattern as to what made certain features easy or challenging to detect, with both groups containing negation, temporal aspects, and features with varying length in characters.

The case with the highest number of errors (n = 18) contained 31 features to look for. Out of the 18 errors, 10 were FPs, and out of these, 4 features looked for negated terms (*no change in diet, no oral contraceptives, no abdominal surgeries* and *no radiation*). Interestingly, some negated expressions from the PNs were erroneously mapped to these negated features such as *denies eating under cokked [sic] foods* being mapped to *no change in diet*, showing that the model is aware that it needs to look for negation but processing it incorrectly. The remaining eight FNs did not reveal a pattern.

Of all errors, 594 were false positives (FPs) across 166 cases and 396 were false negatives (FNs) across 151 cases. The highest number of FPs per case was 12 (2 cases), with the majority of cases containing one or two FPs per case (34 and 35 cases, respectively). For FNs, the highest number of FNs per case was 9 (1 case), with the majority of cases also containing one or two FNs (48 and 37 cases, respectively).

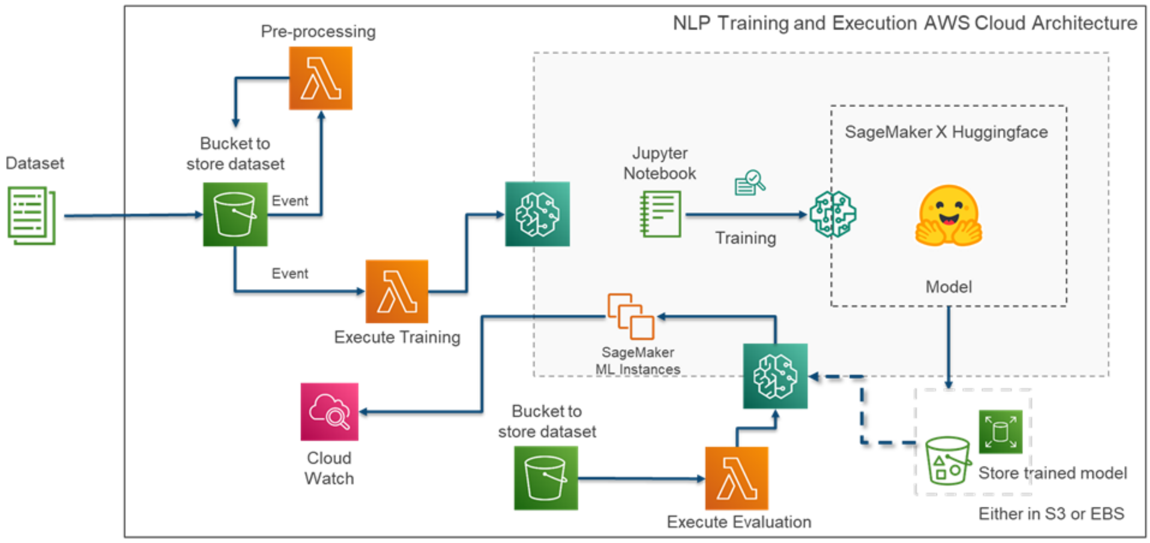


Figure 3: AWS System Architecture. When a new dataset is placed in the S3 bucket, a Lambda function triggers preprocessing and a subsequent Lambda function triggers the training process. Training is performed via SageMaker and Huggingface; final predictions are stored in CloudWatch.

A World CLASSE Student Summary Corpus

Scott A. Crossley[1], Perpetual Baffour[2], Mihai Dascalu[3], Stefan Ruseti[3]
Vanderbilt University[1], The Learning Agency[2], University Politehnica of Bucharest[3]
scott.crossley@vanderbilt.edu, perpetual@the-learning-agency.com, mihai.dascalu@upb.ro, stefan.ruseti@upb.ro

Abstract

This paper introduces the Common Lit Augmented Student Summary Evaluation (CLASSE) corpus. The corpus comprises 11,213 summaries written over six prompts by students in grades 3-12 while using the CommonLit website. Each summary was scored by expert human raters on analytic features related to main points, details, organization, voice, paraphrasing, and language beyond the source text. The human scores were aggregated into two component scores related to content and wording. The final corpus was the focus of a Kaggle competition hosted in late 2022 and completed in 2023 in which over 2,000 teams participated. The paper includes a baseline scoring model for the corpus based on a Large Language Model (Longformer model). The paper also provides an overview of the winning models from the Kaggle competition.

1 Introduction

Many educational applications are interested in assessing student-generated knowledge to assess learning and development. In terms of assessing student comprehension of text, generation effects (Slamecka & Graff, 1978) that result from students writing about what they have read have been shown to substantially improve learning (Bertsch et al., 2007; McCurdy et al., 2020). A number of educational applications have taken advantage of generation effects to enhance students' reading comprehension skills, including Summary Street (Wade-Stein & Kintsch, 2004), the Interactive Strategy Training for Active Reading and Thinking (iSTART) tool (McNamara et al. 2004), the CommonLit online reading program (commonlit.org), and the intelligent Textbooks for Enhanced Lifelong Learning (iTELL) framework (Morris et al., in press).

The most common approach to assessing students' reading comprehension through text generation is likely through text summarization. Text summarization is a valuable tool to build and assess student knowledge (Graham & Harris, 2015; Head et al., 1989) because the process of summarization helps students build and consolidate their knowledge about reading materials (Silva & Limongi, 2019). Text summarization has also been shown to lead to stronger learning gains than other forms of comprehension assessment, including constructed responses (Carroll, 2008), long-form essays (Gil et al., 2010), and traditional assessments like multiple-choice and fill-in-the-blank questions (Mok & Chan, 2016).

While effective, many teachers hesitate to integrate summary assessments of reading in the classroom because manually grading summaries is resource-intensive (Lagakis & Demetriadis, 2021; Li et al., 2018). However, student text summarization can also be assessed automatically through the use of Natural Language Processing (NLP) techniques such as semantic similarity metrics (Crossley et al., 2019; Li et al., 2018; Wade-Stein & Kintsch, 2004) or contextualized word embeddings like those found in Transformer-based language models (Botarleanu et al., 2022; Morris et al., 2023).

To assess student summarization strength automatically, NLP models depend on the availability of large corpora of summaries that have been scored for quality. Unfortunately, previous research has depended on closed-source collections of summaries that are not available to the broader research community (Botarleanu et al., 2022; Crossley et al., 2019; Li et al., 2018; Wade-Stein & Kintsch, 2004), which limits the strength, replication, and generalizability of summarization models. Additionally, many of the corpora used in previous research have included summaries written by crowdsourced workers and not students (Botarleanu et al., 2022; Crossley et al., 2019; Li et al., 2018)

The goal of this study is to introduce the Common Lit Augmented Student Summary Evaluation (CLASSE) corpus. The corpus comprises 11,213 summaries written over six prompts by students in grades 3-12. All summaries were written on the CommonLit website. Each summary was scored by expert human raters on analytic features related to summarization content and wording. The study also introduces a baseline NLP summary scoring model for the corpus as well as the winning models developed in a large-scale data science competition hosted for the corpus.

1.1 Summary writing

Summarizing a reading involves two cognitive processes: comprehension and content production (Li et al., 2018). The reading process leads to the reader's comprehension of the source material. This process generally consists of readers identifying the text's main themes, the ideas that support these themes, and the structures and organization of the text (Spirgel & Delaney, 2016). After reading, summarization allows the student to reproduce the content of the source text that they read and involves the reader (now the writer) generalizing the main ideas contained in the text, synthesizing those ideas, organizing those ideas coherently within the summary, and selecting the proper words and sentence structures to represent the ideas (Brown & Day, 1983; van Dijk & Kintsch, 1983; Galbraith & Baaijen, 2018; León et al. 2006; Nelson & King, 2022). The cognitive demands entailed in summarizing help consolidate the knowledge gained from reading into long-term memory (Silva & Limongi, 2019).

Research indicates that reading to writing tasks like summarization can increase learning outcomes in various content domains (Graham et al., 2020; Silva & Limongi, 2019) and for different types of learners (Rogevich & Perin, 2008; Trabasso & Bouchard, 2002; Shokrpour et al., 2013). A meta-analysis of 56 experiments on the effect of reading on writing tasks found an average weighted effect size of Hedges's $g = 0.3$ ($p < .005$) between pre- and post-tests for students (Silva & Limongi, 2019). Additionally, compared to other methods to assess reading comprehension and knowledge development, like constructed responses, essays, and multiple-choice questions, research has found that summarizations are more effective (Carroll, 2008; Gil et al., 2010; Mok & Chan, 2016).

1.2 Automatic summary evaluation

Despite the effectiveness of having students summarize what they have read, providing feedback to students about the quality of summaries is time-consuming for educators (Gamage et al., 2021; Lagakis & Demetriadis, 2021; Li et al., 2018), thus making human-driven summary assessments difficult to scale.

Noting the importance of summarization in educational settings and the challenges of integrating it into the classroom, researchers have investigated the potential for automatic summary evaluation (ASE) to provide students with computational-derived feedback.

Initial methods for ASE predominantly involved assessing a student's summarization work by comparing it with model summaries crafted by experts. These methods have the advantage of relying on a single expert-derived summary to establish a benchmark for quality. Metrics like ROUGE (Lin & Hovy, 2003) were utilized to assign scores to summaries by examining the frequency of shared words and phrases between the student and expert summaries. Although ROUGE metrics align with the quality ratings given by experts and have been widely adopted in developing summarization tools (Ganesan, 2018; Scialom et al., 2019), the metrics tend to favor basic lexical attributes. This shortcoming can be overcome by employing more sophisticated NLP techniques, such as those involving word embeddings (Ng & Abrecht, 2015).

The earliest attempt at using a word embedding approach to score summaries was likely with the educational application Summary Street. Summary Street allowed students to produce multiple summary drafts and provided feedback to students based on Latent Semantic Analysis (LSA), an early word embedding model. Summary Street used LSA to uncover typical sentences in each section of a text. These sentences were then combined to form a typical summary. Semantic similarity between a student's summary and the typical summary was used to provide feedback to the student about the quality of their summary (Wade-Stein & Kintsch, 2004).

Li et al. (2018) also used LSA to provide scores for summaries written by crowdsourced workers on Mechanical Turk. The crowdsourced summaries were scored by graduate students on four criteria: thesis statement, content, mechanics and grammar, and signal words. Li et al. found that crowdsourced summaries were scored as well as summaries produced by experts using LSA. Li et al. argued that crowdsourced workers could produce a model summary similar to the model summaries produced by experts, which could make it easier to develop model summaries for automated scoring.

Other summarization scoring models have combined more advanced word embedding models and other NLP features to predict quality. For instance, Crossley et al. (2019) developed a summarization model to predict ratings of main idea integration in summaries collected on Mechanical Turk using lexical diversity features, a word frequency metric, and Word2vec semantic similarity scores between summaries and the corresponding source material. The model explained 53% of the variance in ratings.

With the rise of Transformer-based language models, new methods of automated summary evaluation have been evaluated. For instance, Botarleanu et al. (2022) used the summaries of Crossley et al. (2019) to train a Longformer model (Beltagy et al., 2020) to predict overall summarization scores derived from an analytic rubric; their model explained ~55% of the score variance. Morris et al. (in press) used an extended dataset of the one used by Crossley et al. (2019). In addition to crowdsourced summaries, the extended dataset also included summaries written by high school and university students. Morris et al. used the dataset to predict two aspects of summarization quality: content and wording. Using a Longformer, they explained .82 of the variance in the content scores and .70 of the variance in the wording scores.

2 The CLASSE Corpus

While research ASE has gained traction and shown improvements over the last 20 years, the work is somewhat fragmented. A major reason for this is that researchers do not have a large-scale open-source summarization corpus to develop, test, and validate ASE models. Other reasons include the use of different NLP approaches to model summarization quality, the sampling of different populations of writers, and the use of different scoring metrics.

The Common Lit Augmented Student Summary Evaluation (CLASSE) corpus is meant to help address this fragmentation by providing researchers with a gold-standard corpus of open-source summaries written by students. The corpus is freely available in the following repository: https://github.com/scrosseye/CLASSE.

2.1 Summaries

The corpus of summaries found in CLASSE was provided by CommonLit, an online content library and writing platform. The initial corpus comprised 11,353 summaries. Within the CommonLit interface, students read texts and write summaries on those texts. Students also have the opportunity to write essay responses, complete vocabulary quizzes, and answer multiple-choice questions about the text. The final CLASSE corpus after pruning (see section 2.2) comprises 11,213 summaries written over six prompts by students in grades 3-12.

Grade	N	Length (M)	Length (SD)
3	2	172.00	49.50
4	12	77.92	49.19
5	248	87.51	70.17
6	1072	82.58	57.61
7	1177	78.92	58.66
8	1844	76.30	46.06
9	2531	71.62	43.82
10	2247	75.92	50.73
11	1942	73.61	51.15
12	138	80.86	57.22

Table 1: Grade Level

Prompt	N	Length (M)	Length (SD)
Third-Wave	1103	73.88	47.31
Tragedies	2057	63.87	44.93
Jungle	1996	80.52	56.16
Greek	2021	73.72	38.31
Egyptian	2009	85.71	62.58
Nature Nurture	2027	77.10	48.67

Table 2: Prompt Information

The majority of the summaries were written by students in the 6th to 11th grade, with smaller numbers of 3rd, 4th, 5th, and 12th grade students (see

Table 1 for details). English language learning (ELL) status is also available for the students (n = 661). The six prompts were related to the topics of the third wave, poetic tragedies, the novel *The Jungle*, Greek society, Egyptian Society, and the nature/nurture debate (see Table 2 for details). The mean length of the summaries was 75.90 (SD = 50.94, min = 22, max = 651). Text length by grade and prompt is reported in Tables 1 and 2. No demographic information beyond grade and ELL status is available for the students.

2.2 Summary scoring

Summaries were scored by expert raters using a standardized scoring rubric and procedure. An outside agency specialized in providing performance assessment scoring services was hired to score the summaries and initial selection of summaries. Two expert raters scored each summary using a 0-4 scaled analytic rubric to score six criteria important in understanding the quality of summarizations. The rubric was developed based on research into language elements related to essay quality reported by Taylor (2013) and Westby et al. (2010). The initial rubric was revised based on feedback from a panel of teachers and a panel of researchers who specialize in the teaching of summaries. The finalized rubric included analytic ratings for main point/gist (did the summary contain the ideas of the source text), details (did the summary contain all the main ideas of the source text), organization (were the ideas logically presented and linked to each other to support comprehension), voice (was language impartial and objective in the summary), word/paraphrasing (did the summary appropriately paraphrase the source text), and language beyond the source text (did the summary show a range of lexical and syntactic features). The scoring rubric is available at this link. Raters also flagged any summaries that included offensive or emotionally charged language or personally identifiable information (PII). While no PII was reported, 127 summaries were removed for language use.

Raters were provided with ground truth example summaries that had been previously scored. As well, raters went through extensive norming prior to independent rating. After norming, each summary was read by at least two raters and, in some cases, three raters (if there was substantial disagreement). Ratings were conducted by prompt, and rater final scores were averaged such that scores of 3 and 2 were averaged to 2.5.

Score distributions were generally normal except for the details, organization, and wording items, which were positively skewed, indicating a greater number of 1s than 1.5s. Strong correlations were reported among the analytic items, with the highest correlation between organization and voice and the lowest correlation between detail and word (see Figure 1 for a correlation heat map). The exact agreement among analytic items hovered around 70% (see Table 3 for details). Quadratic weighted kappa (QWK) scores for inter-rater reliability were substantial (QWK < .60) for all items except wording, which reported a moderate QWK = .532 (see Table 4).

Significant differences were noted between ELL students and non-ELL students for both content scores (t = 3.993, p < .001) and wording scores (t = 5.684, p < .001). Descriptive statistics for content and wording scores by ELL and non-ELL students are reported in Table 5. No significant correlations were reported between grade level and content score (r = -0.036, p > .050) and wording scores (r = -0.049, p > .050). Descriptive statistics for content and wording scores by grade are reported in Table 6.

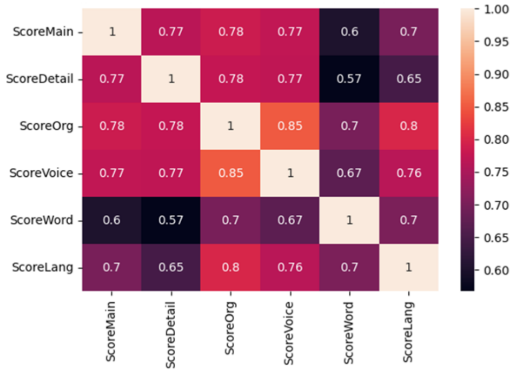


Figure 1: Heatmap for correlations among analytic item scores

Item	Adjacent Low	Exact	Adjacent High
Main Idea	13.2	73.0	13.2
Details	13.9	72.0	13.9
Organization	15.1	69.0	15.1
Voice	15.4	69.0	15.4
Wording	16.9	65.0	16.9
Language	11.8	76.0	11.8

Table 3: Exact and adjacent percentages

Item	**QWK**
Main Idea	0.617
Details	0.673
Organization	0.694
Voice	0.683
Wording	0.532
Language	0.653

Table 4: Quadratic Weighted Kappa (QWK) for inter-rater reliability

Group	Content M (SD)	Wording M (SD)
Non-ELL	0.016 (1.002)	0.023 (0.999)
ELL	-0.136 (0.950)	-0.186 (0.910)

Table 5: Descriptive statistics for content and wording scores for ELL and non-ELL students

Grade	Content M (SD)	Wording M (SD)
3	1.593 (2.015)	1.041 (1.419)
4	-0.201 (1.131)	-0.359 (0.759)
5	-0.056 (1.115)	-0.14 (0.964)
6	0.036 (1.067)	-0.039 (0.939)
7	-0.063 (1.054)	-0.071 (0.953)
8	0.008 (0.985)	0.076 (0.963)
9	0.025 (0.923)	0.098 (0.955)
10	0.081 (1.01)	0.084 (1.057)
11	-0.061 (1.002)	-0.142 (1.04)
12	-0.073 (1.008)	-0.146 (0.967)

Table 6: Descriptive statistics for content and wording scores by grade

2.3 Dimensionality reduction

Since the rubric consisted of six criteria, many of which were related, we conducted a Principal Component Analysis (PCA) to assess the potential to reduce the dimensionality of the six analytic scores into a smaller number of related constructs.

Before conducting the PCA, the human scores were standardized using z-score normalization. An initial PCA was performed with all possible factors (n = 6). A Kaiser-Meyer-Olkin (KMO) measure of sampling adequacy indicated that no variables need to be removed (i.e., all KMO values were above .5), and the overall KMO score = .918 indicated a “meritorious” sample (Kaiser & Rice, 1974). The PCA reported a Bartlett’s test of sphericity, $\chi^2 = 61{,}533.87$, $p < .001$, indicating that correlations between the analytic scores were sufficiently large for the PCA. Within the components, there was a break in the cumulative variance explained between the second and the third components. Considering this break, we decided on a 2-component solution when developing the PCA. These 2 components explained approximately 86% of the shared variance in the data from the initial PCA.

The first component was related to *content* (i.e., Component 1), and the analytic items details, main point, voice, and organization were combined into a weighted score. The analytic items wording/paraphrasing and language beyond the source were combined into a weighted score designated as *wording* (i.e., Component 2). The component scores were z-score normalized and rescaled such that zero represents the mean for each principal component, and one unit represents one standard deviation.

2.4 Final dataset

The final dataset comprises 11,213 summaries and metadata in tabular format and is available at this link. The dataset contains student ID numbers (anonymous), the prompt ID for each summary, the text of the summary, the average content and wording scores for the summary, the student grade level, and ELL classification, along with the data split that was used in the Kaggle competition (see section 4 for details). The data was split into a training set (n = 7,165), a validation set used as a test set for the public leaderboard on Kaggle (n = 2,021), and a test set used for the private leaderboard on Kaggle (n = 2,027). The splits were selected so that the difference in scores across the splits was similar to demographic information (grade and ELL classification). The training set comprised four prompts (Third Wave, Tragedies, The Jungle, and Egyptian Society). The validation set included a single prompt (Greek Society), as did the test set (Nature versus Nurture).

3 Baseline prediction model for CLASSE corpus

We developed a simple baseline model for the CLASSE by finetuning a Longformer model (Beltagy et al., 2020) to predict the content and wording scores, given the original text and the summary. The baseline model is not meant to extend the technical boundaries of summary classification models but rather provide a simple metric from which to measure scoring gains.

3.1 Model description

An encoder architecture was chosen for the baseline model over a decoder model because the prediction task is a regression that involves continuous values. Since a decoder model is used to generate text, the output values would have to be expressed in words. This does not imply that a decoder cannot be used for this task, but an encoder model seemed a better fit for the data.

The input for the model consisted of both the summary and the source text, separated by the “sep” token. Given the length of the input exceeding 512 tokens, a Longformer model was chosen as a baseline encoder.

Several options were tested for the final summary embedding: pooled output, average of all tokens, and average of summary tokens. Adding a hidden layer between the embedding and the decision layer was also considered. The best configuration used the average of the summary tokens followed by a dropout layer of 20%, no hidden layer or output activation, and a learning rate of 1e-5 using the Adam optimizer. The mean squared error sum for the two tasks was used as a loss function. The lowest validation loss was obtained after three epochs, and the corresponding model was used for evaluation. The model was trained on the training set, validated on the validation set, and tested on the test set used in the Kaggle competition.

3.2 Prediction performance

The metric used for the Kaggle competition was Mean Columnwise Root Mean Squared Error (MCRMSE), which is the average of the RMSE for the two scoring components (content and wording). RMSE is a general error metric used for numerical predictions that punishes large errors in predictions. An RMSE score of zero represents a perfect fit between the model and the outcome variables (in this case, content and wording scores). Thus, a lower RMSE represents a better model.

The results for the baseline model for each partition, each component, and the average scores are presented in Table 6. The model performed well on the training and validation sets for content, but it performed less accurately on the wording scores. Model performance dipped for the content scores in the test set and fell for the wording scores. The overall scores for MCRMSE were strong for the training set but fell in the validation and test sets. The final MCRMSE reported for the test set was 0.582.

Partition	Content RMSE	Wording RMSE	MCRMSE
Train	0.375	0.427	0.401
Validation	0.415	0.614	0.515
Test	0.480	0.683	0.582

Table 6: Baseline model performance

4 Kaggle Competition

The CLASSE dataset was the subject of a recently completed Kaggle competition (CommonLit - Evaluate Student Summaries). The goal of the competition was for data scientists to assess the quality of summaries in the CLASSE corpus in terms of content and wording. The winning models provide state-of-the-art techniques for modeling summary scoring in student data and demonstrate the potential for the CLASSE corpus to inform student learning and interventions.

The competition started in July of 2023 and ended in October of 2023. Over 2,000 teams comprising ~2,500 competitors entered the competition, creating over 40,000 summary scoring models. All winning models are freely available for use through an MIT license and provided on the Kaggle website. The Kaggle website also provides the training and validation data used in the competition.

5 Kaggle competition results

As mentioned earlier, success in the Kaggle competition was demonstrated through a model’s mean column-wise root mean squared error (MCRMSE), which represented the average Root Mean Squared Error (RMSE) across the content and wording scores.

The top 17 teams reported an MCRMSE below .46, with the first-place team reporting an MCRMSE of .452. These models thus outperformed our baseline model (MCRMSE = 0.582). Within the top five entrants, the most common approach used when modeling the summary scores was an ensemble model using the DeBERTa encoder. This approach was used with the second through fifth place teams, with all teams except the fifth place team using only DeBERTa models (the fifth place team used DeBERTa v3 large and a LightGBM ensemble model). The first-place team used a single

DeBERTa model (v3 large), but critically, they augmented the training set by creating 1000 new prompts with associated sources using generative AI. For each prompt, they also created 21 summaries and pseudo-labeled those summaries. Other common approaches used to improve the models included using a head mask for only the student summaries instead of a normal attention mask, using generative AI models to generate varieties of the existing prompts, hyperparameter searches, extending the inference max length, and using all of the input (summary, prompt, source, and title) in the training models.

6 Discussion and conclusion

This paper has introduced the CLASSE corpus, the scoring metrics for the corpus, and a baseline model for summary scoring based on a DeBERTa Transformer-based encoder. The paper also introduced the winning summarization models from the Kaggle competition held in support of the CLASSE corpus.

The CLASSE comprises 11,213 summaries written over six prompts by students in grades 3-12 while using the CommonLit website. Each summary was scored by expert human raters on analytic features related to summarization content and wording.

Reliability metrics for the human scoring indicated substantial reliability in all items except paraphrasing/wording, which reported moderate reliability. Paraphrasing is the restatement of a passage such that the propositional meaning is similar, but the words and structures differ. Recognizing when words differ between passages is relatively easy, but recognizing the alteration of clauses is a difficult task (Barzilay & Lee, 2003), which may explain the moderate reliability reported by human raters.

The analytic scores were aggregated into components using a principal component analysis (PCA) to better represent the underlying structure of the human ratings. The PCA reported two components related to content and wording. Content included features related to main ideas, details for those ideas, the organization of those ideas, and the objectivity of how those ideas were presented. The content component provides an overall assessment of how the ideas in the source text are distilled into a coherent and objective framework in the student summaries. Wording includes features related to paraphrasing and the use of language beyond the source. This component was concerned with the manner in which the summary presented the ideas from the source text, specifically, did the summary use original wording (paraphrasing) and whether this wording was lexically and syntactically complex.

The baseline model introduced in this paper used a Longformer model that used both the summary and the source text as input for model predictions. The Longformer performed well on the training data but reported drops in the validation and test data. This is the result of the Longformer model learning the patterns of successful summarization specific to the four prompts in the training set but not learning how to extend scoring beyond those prompts to the two unique prompts in the validation and test sets.

The results of the subsequent Kaggle competition showed a number of innovations that helped competitors produce winning models, many of which addressed the limitations of the baseline model. The winning model used a single Transformer encoder (DeBERTa v3 large), but, importantly, they augmented their training data to include a much larger number of prompts and summaries written on those prompts. Extending the number of prompts and summaries allowed the model to generalize better to the unique prompts found in the validation and test set. Other innovations in summary scoring that resulted from the Kaggle competition included pseudo-labeling of AI generated summaries for content and wording scores, the use of head masks, and extending the inference max length.

6.1 Limitations

While the CLASSE corpus is the largest corpus of student summaries, with individual human scores assigned to each summary, there are limitations to the corpus. An important limitation is that there are only six source texts and prompts for the corpus. As noted, the first-place solution on Kaggle augmented the CLASSE dataset by creating 1,000 new prompts and source text along with pseudo-labeling these summaries, all of which are available in the winning model. However, augmenting data is different from collecting real data, and future developments of CLASSE or newer summarization datasets should include a greater number of prompts.

Another limitation of the CLASSE corpus is that certain grades (i.e., 6th-11th grades) were over-represented in the corpus. Greater representation of lower and upper grades, including college-level students, is warranted. Finally, while the CLASSE corpus includes some individual difference metrics, little information is known about the writers in terms of gender, race/ethnicity, or socio-economic status, all of which are important student-oriented variables that may influence human ratings.

6.2 Future directions

The goals of the Kaggle competition were to publicize and make freely available a large-scale corpus of student-written summaries and advanced models of assessing summarization quality. Future directions include integrating the models developed in the Kaggle competition into educational applications to help students receive feedback on summaries written within these applications. Knowing the strength of generation effects on learning (Bertsch et al., 2007; McCurdy et al., 2020) and the strengths of summarization tasks in general (Carroll, 2008; Gil et al., 2010; Mok & Chan, 2016), the integration of CLASSE corpus scoring models into educational applications will ensure students quickly receive formative feedback about their summaries, allowing for deliberative practice during the revision process and increased learning.

Acknowledgments

The authors would like to thank Michelle Brown at commonlit.org along with Ulrich Boser, Meg Benner, and Alex Franklin at the Learning Agency for their help in releasing the corpus and hosting the Kaggle competition.

References

Barzilay, R., & Lee, L. 2003. Learning to paraphrase: an unsupervised Approach Using Multiple-Sequence Alignment. *Proceedings of HLT-NAACL*. 16-23. Edmonton, Canada.

Beltagy, I., Peters, M. E., & Cohan, A. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Bertsch, S., Pesta, B. J., Wiscott, R., & McDaniel, M. A. 2007. The generation effect: A meta-analytic review. *Memory & cognition, 35*, 201-210.

Brown, A. L., & Day, J. D. 1983. Macrorules for summarizing texts: The development of expertise. *Journal of Verbal Learning and Verbal Behavior, 22*, 1–14.

Botarleanu, R.-M., Dascalu, M., Allen, L. K., Crossley, S. A., & McNamara, D. S. 2022. Multitask Summary Scoring with Longformers. In M. M. Rodrigo, N. Matsuda, A. I. Cristea, & V. Dimitrova (Eds.), *Artificial Intelligence in Education* (Vol. 13355, pp. 756–761). Springer International Publishing. https://doi.org/10.1007/978-3-031-11644-5_79

Crossley, S. A., Kim, M., Allen, L., & McNamara, D. 2019. Automated summarization evaluation (ASE) using natural language processing tools. In *Artificial Intelligence in Education: 20th International Conference, AIED 2019, Chicago, IL, USA, June 25-29, 2019, Proceedings, Part I 20* (pp. 84-95). Springer International Publishing.

Carroll, D.W. 2008. Brief report: A simple stimulus for student writing and learning in the introductory psychology course. *North American Journal of Psychology, 10*, 159–164.

Galbraith, D., & Baaijen, V. M. 2018. The Work of Writing: Raiding the Inarticulate. *Educational Psychologist*, *53*(4), 238–257. https://doi.org/10.1080/00461520.2018.1505515

Gamage, D., Staubitz, T., & Whiting, M. 2021. Peer assessment in MOOCs: Systematic literature review. *Distance Education*, *42*(2), 268–289. https://doi.org/10.1080/01587919.2021.1911626

Ganesan, K. 2018. *ROUGE 2.0: Updated and Improved Measures for Evaluation of Summarization Tasks*. https://doi.org/10.48550/ARXIV.1803.01937

Gil, L., Bråten, I., Vidal-Abarca, E., & Strømsø, H. I. 2010. Summary versus argument tasks when working with multiple documents: Which is better for whom? *Contemporary Educational Psychology, 35,* 157–173.

Graham, S., & Harris, K. R. 2015. Common Core State Standards and Writing: Introduction to the Special Issue. *The Elementary School Journal*, *115*(4), 457–463. https://doi.org/10.1086/681963

Graham, S., Kiuhara, S. A., & MacKay, M. 2020. The Effects of Writing on Learning in Science, Social Studies, and Mathematics: A Meta-Analysis. *Review of Educational Research*, *90*(2), 179–226. https://doi.org/10.3102/0034654320914744

Head, M. H., Readence, J. E., & Buss, R. R. 1989. An examination of summary writing as a measure of reading comprehension. *Reading Research and Instruction*, *28*(4), 1–11. https://doi.org/10.1080/19388078909557982

Kaiser, H. F., & Rice, J. 1974. Little jiffy, mark IV. *Educational and psychological measurement*, *34*(1), 111-117.

Lagakis, P., & Demetriadis, S. 2021. Automated essay scoring: A review of the field. *2021 International*

Conference on Computer, Information and Telecommunication Systems (CITS), 1–6. https://doi.org/10.1109/CITS52676.2021.9618476

León, J. A., Olmos, R., Escudero, I., Cañas, J. J., & Salmerón, L. 2006. Assessing short summaries with human judgments procedure and latent semantic analysis in narrative and expository texts. *Behavior Research Methods, 38*, 616–627. https://doi.org/10.3758/BF03193894

Li, H., Cai, Z., & Graesser, A. C. 2018. Computerized summary scoring: Crowdsourcing-based latent semantic analysis. *Behavior Research Methods*, *50*(5), 2144–2161. https://doi.org/10.3758/s13428-017-0982-7

Lin, C.-Y., & Hovy, E. 2003. Automatic evaluation of summaries using N-gram co-occurrence statistics. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology NAACL'03*, *1*, 71–78. https://doi.org/10.3115/1073445.1073465

McCurdy, M. P., Viechtbauer, W., Sklenar, A. M., Frankenstein, A. N., & Leshikar, E. D. 2020. Theories of the generation effect and the impact of generation constraint: A meta-analytic review. *Psychonomic Bulletin & Review*, *27*(6), 1139–1165. https://doi.org/10.3758/s13423-020-01762-3

McNamara, D. S., Levinstein, I. B., & Boonthum, C. 2004. iSTART: Interactive strategy training for active reading and thinking. *Behavior Research Methods, Instruments, & Computers*, *36*(2), 222-233.

Mok, W. S. Y., & Chan, W. W. L. 2016. How do tests and summary writing tasks enhance long-term retention of students with different levels of test anxiety? *Instructional Science, 44*, 567–581.

Morris, W., Crossley, S., Holmes, L., Ou, C., McNamara, D., Dascalu, M. 2023 Using Transformer Language Models to Provide Formative Feedback in Intelligent Textbooks. In N. Wang, G. Rebolledo-Mendez, V. Dimitrova, N. Matsuda, & O. C. Santos (Eds.), *Artificial Intelligence in Education (AIED)* (pp. 484–489). Springer Nature Switzerland.

Morris, W., Crossley, S., Holmes, L., Ou, Chaohua, Dascalu, M., & McNamara, D. in press. Formative Feedback on Student-Authored Summaries in Intelligent Textbooks using Large Language Models. *Journal of Artificial Intelligence in Education*.

Nelson, N., & King, J. R. 2022. Discourse synthesis: Textual transformations in writing from sources. *Reading and Writing*. https://doi.org/10.1007/s11145-021-10243-5

Ng, J.-P., & Abrecht, V. 2015. *Better Summarization Evaluation with Word Embeddings for ROUGE* (arXiv:1508.06034). arXiv. http://arxiv.org/abs/1508.06034

Rogevich, M., & Perin, D. 2008. Effects on science summarization of a reading comprehension intervention for adolescents with behavior and attention disorders. *Exceptional Children, 74*, 135–154.

Scialom, T., Lamprier, S., Piwowarski, B., & Staiano, J. 2019. *Answers Unite! Unsupervised Metrics for Reinforced Summarization Models*. https://doi.org/10.48550/ARXIV.1909.01610

Shokrpour, N., Sadeghi, A., & Seddigh, F. 2013. The effect of summary writing as a critical reading strategy on reading comprehension of Iranian EFL learners. *Journal of Studies in Education, 3,* 127–138. https://doi.org/10.5296/jse.v3i2.2644

Silva, A., & Limongi, R. 2019. Writing to Learn Increases Long-term Memory Consolidation: A Mental-chronometry and Computational-modeling Study of "Epistemic Writing." *Journal of Writing Research*, *11*(vol. 11 issue 1), 211–243. https://doi.org/10.17239/jowr-2019.11.01.07

Slamecka, N. J., & Graf, P. 1978. The generation effect: Delineation of a phenomenon. *Journal of Experimental Psychology: Human Learning and Memory*, *4* (6), 592–604. https://doi.org/10.1037/0278-7393.4.6.592

Spirgel, A. S., & Delaney, P. F. 2016. Does writing summaries improve memory for text? *Educational Psychology Review, 28*, 171–196.

Taylor, D. M. 2013. Writing rubrics as formative assessments in an elementary classroom. *Education and Human Development Master's Theses, Paper*, *258*.

Trabasso, T., & Bouchard, E. 2002. Teaching readers how to comprehendtexts strategically. In C. Block & M. Pressley (Eds.), *Comprehension instruction: Research-based best practices* (pp. 176–200). New York, NY: Guilford Press.

van Dijk, T. A., & Kintsch, W. 1983. *Strategies of discourse comprehension* (pp. 11–12). New York, NY: Academic Press.

Wade-Stein, D., & Kintsch, E. 2004 Summary Street: Interactive computer support for writing. *Cognition and Instruction, 22*, 333–362.

Westby, C., Culatta, B., Lawrence, B., & Hall-Kenyon, K. 2010. Summarizing expository texts. *Topics in Language Disorders, 30*, 275–287.

Improving Socratic Question Generation using Data Augmentation and Preference Optimization

Nischal Ashok Kumar and **Andrew Lan**
University of Massachusetts Amherst
{nashokkumar, andrewlan}@cs.umass.edu

Abstract

The Socratic method is a way of guiding students toward solving a problem independently without directly revealing the solution to the problem by asking incremental questions. Although this method has been shown to significantly improve student learning outcomes, it remains a complex labor-intensive task for instructors. Large language models (LLMs) can be used to augment human effort by automatically generating Socratic questions for students. However, existing methods that involve prompting these LLMs sometimes produce invalid outputs, e.g., those that directly reveal the solution to the problem or provide irrelevant or premature questions. To alleviate this problem, inspired by reinforcement learning with AI feedback (RLAIF), we first propose a data augmentation method to enrich existing Socratic questioning datasets with questions that are invalid in specific ways. Also, we propose a method to optimize open-source LLMs such as LLama 2 to prefer ground-truth questions over generated invalid ones, using direct preference optimization (DPO). Our experiments on a Socratic questions dataset for student code debugging show that a DPO-optimized LLama 2-7B model can effectively avoid generating invalid questions, and as a result, outperforms existing state-of-the-art prompting methods[1].

1 Introduction

Learning based on a conversation that consists of questions and answers, where the student responds to questions posed by a more knowledgeable instructor, has been proven to be effective in teaching students about a particular concept (Wood et al., 1976). In particular, *Socratic questioning*, which refers to a way for the instructor to guide a student to solve a problem (within their zone of proximal development) by asking them questions that promote thinking while not directly revealing the solution (Quintana et al., 2018), is a very effective pedagogical method in conversation-based learning and tutoring.

Recent advances in large language models (LLMs) (Bubeck et al., 2023) have led to the rapid development of chatbots that promote student learning by automatically generating the instructor's utterances (Dan et al., 2023; Kazemitabaar et al., 2024; Tanwar et al., 2024). One key area of interest in the development of such chatbots is question generation, which can help students solve logical problems in the mathematics and programming domains (Al-Hossami et al., 2023; Shridhar et al., 2022). Typically, question generation in educational applications has focused on generating practice or assessment questions, in biology exams (Wang et al., 2018), reading comprehension (Ashok Kumar et al., 2023), math practice (Wang et al., 2021), and programming exercises (Sarsa et al., 2022). As a specific form of question generation, Socratic question generation has gained attention, owing to its effectiveness in improving student learning outcomes by eliciting critical thinking and self-discovery during problem-solving (Paul and Elder, 2007).

Socratic questions generation is a complex task because it involves mapping out the step-by-step thought process of students during problem-solving, locating the cause of their error, and providing effective questions without revealing the solution. Manually generating Socratic questions can be a cognitively demanding and time-consuming task for instructors. Several recent works proposed to automatically generate Socratic questions using LLMs: In math education, (Shridhar et al., 2022) shows that generating a sequence of Socratic subquestions and prompting students to answer helps them solve math word problems more successfully. In computer science education, (Al-Hossami et al., 2024, 2023) releases a dataset on Socratic questions for student code debugging and provides baselines

[1] The code for our paper can be found at: `https://github.com/umass-ml4ed/socratic-quest-gen`

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 108–118
June 20, 2024 ©2024 Association for Computational Linguistics

based on LLM prompting and finetuning. In particular, the authors prompt GPT-3.5-turbo and GPT-4 (Bubeck et al., 2023) in a chain-of-thought manner (Wei et al., 2022) to generate Socratic questions. A human study by the authors shows that the generated questions can sometimes be invalid in several different ways, including being irrelevant to the problem, repetitive of earlier dialogue turns, or too direct and revealing the solution prematurely, which may hamper students' learning processes. Since GPT models are proprietary and expensive, the authors also attempt to fine-tune the open-source Flan-T5 model (Chung et al., 2022); however, doing so proves to be ineffective due to its insufficient scale and the pretraining procedure used.

In this paper, we propose a method to improve the validity of automatically generated Socratic questions using open-source LLMs. Our method is inspired by recent developments in reinforcement learning with AI feedback (RLAIF) (Lee et al., 2023); our method consists of two phases, data augmentation and preference optimization. Specifically, our contributions are as follows:

- To the best of our knowledge, this work is the first to introduce a data augmentation method to create negative samples, i.e., invalid questions, to help us train LLM-based Socratic question generation methods.
- We use the preference information in the dataset, i.e., pairs of valid and invalid Socratic questions, to optimize Llama 2 (Touvron et al., 2023), an open-source LLM, using direct preference optimization (DPO). (Rafailov et al., 2023).
- We show that our method using the Llama 2-7B model outperforms existing state-of-the-art methods that rely on larger, proprietary models such as GPT-3.5 and GPT-4 on the Rouge-L metric and are comparable in terms of BERTScore. We also use a series of case studies to illustrate the quality of Socratic questions we generate and that DPO consistently outperforms supervised fine-tuning (SFT).

2 Related Work

2.1 Question Generation in Education

In education, question-generation systems are used to create learning materials and problem sets for quizzes and exams. (Wang et al., 2021) introduces a framework for generating math word problems that incorporates a module for checking the consistency of the word problem generated in terms of the underlying equations that it solves. Our idea of checking the consistency of the synthetically generated samples in data augmentation is inspired by theirs. (Ashok Kumar et al., 2023) proposes a data augmentation and an over-generate and rank method to fine-tune a language model Flan-T5 (Chung et al., 2022) to generate questions for reading comprehension. Their data augmentation method prompts a larger LLM to augment the dataset with valid questions (positive examples) corresponding to a passage in the reading comprehension and then uses this augmented dataset for standard fine-tuning of a smaller open-source LLM. Unlike their work, our data augmentation method involves prompting a larger LLM to generate invalid questions (negative examples) to create a preference dataset that we use for performing preference optimization on a smaller open-source LLM. In computer science education, recent works show the effectiveness of LLMs like OpenAI Codex and GPT-4 (Sarsa et al., 2022; Kumar and Lan, 2024) on generating programming exercise questions, code explanations, and test cases. (Al-Hossami et al., 2024, 2023) introduce a Socratic code debugging dataset, to help a student debug their code along with maximizing the students' learning outcomes. Their experiments with prompting models like GPT-3.5-turbo, and GPT-4 show that these models tend to hallucinate and produce invalid questions. To address this issue, our work builds upon theirs to fine-tune language models to align the generated questions towards ground-truth human preferences and discourage the models from generating invalid questions.

2.2 Reward/ Preference Optimization

Fine-tuning language models to align with human preferences has proven to be beneficial in various natural language processing tasks (Kreutzer et al., 2018; Stiennon et al., 2020; Ziegler et al., 2019; Ouyang et al., 2022). Traditional methods first learn a reward model using a dataset of human preferences and optimize the language model for the downstream task using the rewards obtained from the reward model with reinforcement learning (RL) algorithms such as PPO (Schulman et al., 2017). There are two drawbacks to this method. First, it is hard to obtain a dataset of human preferences as it is an expensive and sometimes cognitively demanding task. To address this issue, RLAIF procures rewards from an AI system, such as an LLM, and has become a scalable and cheaper alternative (Lee et al., 2023). Second, although preference optimization of LLMs using RL algorithms like PPO is effective, it is significantly more challenging and time-consuming than traditional supervised learning as it involves tuning multiple LLMs and sampling rewards in real time. To address this issue, the DPO method (Rafailov et al., 2023) optimizes a language model to a preference dataset in an RL-free manner by formulating the problem as a binary classification task.

In the domain of education, (Shridhar et al., 2022) proposes a reward-based method to generate Socratic sub-questions to solve math word problems. Similar to our method they define reward characteristics like fluency, granularity, and answerability to prefer sub-questions that have these desired characteristics. They use REINFORCE (Williams, 1992) a popular RL algorithm to optimize their model by sampling rewards from external systems in real time. Our method is different from theirs as we first prompt an LLM to generate invalid Socratic questions (negative examples) to construct a preference dataset. We then use this fixed dataset to tune an open-source LLM in an RL-free method, i.e., using DPO which makes the training more stable and less complex. (Hicke et al., 2023) proposes a DPO-based method for fine-tuning LLama 2 (Touvron et al., 2023) for question-answering on a dataset of Piazza posts for an introductory programming course. They create a proxy preference dataset by using the edit history of Piazza posts by preferring the final versions of answers as opposed to the earlier versions. However, the setting of their work is different from ours as we focus on Socratic question generation and propose a method to create the preference dataset using data augmentation. (Scarlatos et al., 2024) propose a method to perform DPO on LLama 2 for the task of feedback generation to help students solve mathematics word problems. To create preference pairs they prompt LLMs like Codex (Chen et al., 2021) and GPT-3.5 turbo to generate bad feedback and rate the feedback based on a pre-defined rubric using GPT-4. Our problem setting is different from theirs as we focus on the programming education domain and for our task the LLM needs to provide a series of step-by-step feedback in the form of a dialogue-based interaction through Socratic questions instead of just providing the feedback once for a given problem.

3 Problem Definition and Dataset

We study the problem of Socratic question generation in conversations between a *Student* and an *Instructor*, where the Instructor's goal is to guide the Student through the process of solving a problem. Concretely, our goal is to generate Socratic questions at a particular dialogue turn for the instructor during the conversation, given the dialogue history and contextual information about the problem the Student is trying to solve and their solution.

In this work, we use the dataset for code debugging introduced in (Al-Hossami et al., 2024, 2023). The dataset is based on didactic conversations between a Student and an Instructor, where the Student is a novice programmer tasked with writing a program for a given problem. The dataset consists of the Student's buggy code submissions along with a dialogue between the Instructor and the Student, where the Instructor asks Socratic questions in the form of a conversation to help the Student debug their code. The conversation consists of dialogue turns with each Instructor utterance being a collection of several possible "ground-truth" Socratic questions at that dialogue turn. The dataset also contains metadata including the problem statement, the test cases, the bug description, and code fixes to resolve the bug. In total, there are 38 problems with more than 50 different bugs in student solutions, and conversations centered around these buggy codes containing more than 1900 dialogue turns. The dataset is split into two subsets, a train set and a test set which contain 135 and 16 dialogues, respectively, spread across different problems.

4 Proposed method

In this section, we describe our method for the task of Socratic question generation. Our method involves two phases: First, data augmentation, and second, preference optimization, as shown in Figure 1.

4.1 Data Augmentation

Inspired by methods in RLAIF (Lee et al., 2023), we augment the dataset with invalid Socratic questions constructed by prompting GPT-4 (Bubeck et al., 2023), which provides realistic negative samples for LLM-based question generation meth-

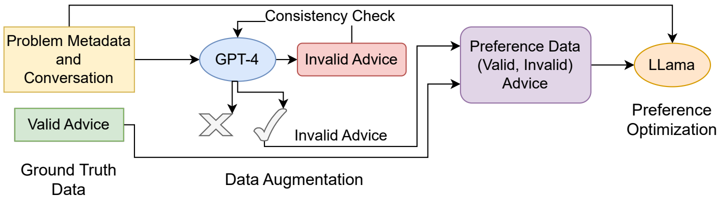


Figure 1: Illustration of our method for LLM-based Socratic question generation, which consists of two phases, data augmentation, and preference optimization.

ods to train on. We follow the method described in (Ashok Kumar et al., 2023) to prompt an LLM to generate synthetic data and employ another instance of the LLM for checking the quality/consistency of the generated synthetic data. Following the definition mentioned in (Al-Hossami et al., 2024), invalid Socratic questions fall into the four following categories:

- **Irrelevant** questions that are not useful for the student, as they shift focus from the actual bug, which may confuse the student.
- **Repeated** questions that have already been asked in previous dialogue turns, which are meaningless to the student.
- **Direct** questions that directly reveal the bug to the student, which do not prompt students to think and may hinder their learning process.
- **Premature** questions which prompt the student to make code fixes before identifying the bug, which may confuse the student.

To generate invalid questions via an LLM, we construct a few-shot prompt that consists of 1) the definition of the categories as mentioned above and 2) an in-context example for each of the invalid question categories detailed above. Our prompt encourages the model to reason using a chain-of-thought method, by first generating the "reasoning process/logic" behind an invalid question, followed by the question (Wei et al., 2022). We generate invalid questions corresponding to all four categories at every dialogue turn where the ground truth is provided.

Following (Ashok Kumar et al., 2023; Wang et al., 2021), we use a consistency checking step where we prompt GPT-4 to check the consistency of the generated questions to filter out inconsistent questions from the augmented dataset. Inconsistent questions are those that do not belong to any of the invalid categories listed above. We pose the consistency checking step as a classification task where GPT-4 predicts a label for each generated question over six categories, including the four invalid categories and two additional categories: "good" and "incorrect". Good questions are acceptable Socratic questions at that particular dialogue turn and cannot be used as negative samples. Incorrect questions are unrelated to the problem and the dialogue itself and are often erroneous due to LLM hallucination, which provides little value as easy-to-tell negative samples. To maintain high data quality of our preference dataset, we discard all samples that are predicted as "good" or "incorrect", to get the final set of synthetically generated invalid questions.

Finally, we construct a preference dataset consisting of 2500 tuples of valid and invalid Socratic questions. In the preference pairs, valid questions are taken from the ground truth questions in the original dataset, while the invalid questions are generated synthetically as described above. Each valid question from the original dataset is paired with every synthetically generated invalid question of all categories to form the augmented dataset.

4.2 Preference Optimization

In this step, we fine-tune an open-source LLM, Llama 2 (Touvron et al., 2023) for Socratic question generation using DPO (Rafailov et al., 2023). The first step is to perform SFT, i.e., we use the original dataset, D, as is to fine-tune LLama 2 for Socratic question generation. For a given conversation in the train set, we first split the dialogue into constituent dialogue turns. The input to LLama 2 is a prompt (p) that consists of a systems message that instructs the LLM to generate a Socratic question, the problem metadata, and the current dialogue history (between the Student and the Instructor). The output is the valid Socratic question (q_v) corresponding to that dialogue turn in the dataset. In the cases where multiple Socratic questions were given for a dialogue turn, we treat each one as a different output associated with the same input for fine-tuning LLama 2. As shown in Equation 1, the simple SFT step learns a reference policy π_{ref} by minimizing the loss $\mathcal{L}_{SFT}$, which serves as the starting point for preference optimization.

The second step is to perform preference optimization where we fine-tune Llama 2 on the preference dataset, D_P, that we obtain from the data augmentation phase, using the same prompt, p, as input that was used for SFT, but with two outputs: the valid question q_{v} and the invalid question q_{iv}, for that dialogue turn. As shown in Equation 2, this preference optimization step learns a human preference-aligned policy π_θ, given the reference policy π_{ref} obtained from Equation 1, by formulating the task as a binary classification task, minimizing the negative log-likelihood loss $\mathcal{L}_{DPO}$, where σ is the Sigmoid function. This minimization leads to learning π_θ, by increasing the likelihood of the valid question and decreasing the likelihood of the invalid question while remaining close to the reference policy π_{ref} which is governed by the hyperparameter β. Here θ is the parameters of the preference-aligned policy which is simply the parameters of the neural network, in our case LLama 2.

$$\mathcal{L}_{\text{SFT}}(\pi_{\text{ref}}) = -\mathbb{E}_{(q_{\text{v}},p)\sim D}[\log \pi_{\text{ref}}(q_{\text{v}}|p)] \qquad (1)$$

$$\begin{aligned}&\mathcal{L}_{DPO}(\pi_\theta;\pi_{\text{ref}}) = \\ &\quad -\mathbb{E}_{(q_{\text{v}},q_{\text{iv}},p)\sim D_P}\left[\log \sigma(\beta \log \frac{\pi_\theta(q_{\text{v}}|p)}{\pi_{\text{ref}}(q_{\text{v}}|p)}\right. \\ &\qquad\qquad \left. - \beta \log \frac{\pi_\theta(q_{\text{iv}}|p)}{\pi_{\text{ref}}(q_{\text{iv}}|p)})\right] \qquad (2)\end{aligned}$$

5 Experimental Settings

In this section, we detail the implementation setup, methods compared, and metrics used to evaluate our Socratic question generation method.

Implementation details. In the data augmentation phase, we query OpenAI's[2] GPT-4 using a rate-based API. We set the temperature of the GPT-4 model to 0.5 to encourage moderate randomness in the outputs. For the consistency checking GPT-4 model, we use a temperate of 0 to maintain determinism. In the preference optimization phase, we use Code-Llama (7B) (Roziere et al., 2023) pre-trained for instruction following tasks, particularly on code data[3]. We load our Code-Llama model in an 8-bit configuration and train using QLora (Dettmers et al., 2023) with the *peft*[4] HuggingFace library to facilitate efficient fine-tuning. For the SFT step, we fine-tune the model for 5 epochs with a learning rate of 3e-5, and a batch size of 2 by accumulating gradients for creating a virtual batch size of 64 which takes about 10 hours to train on a single Nvidia A6000 GPU. For the DPO step, we fine-tune the model for 1 epoch with a learning rate of 3e-5 and a β (which denotes the KL-loss (Joyce, 2011) between the preference policy learned and the reference SFT policy) of 0.1, with a batch size of 2, which takes about 6 hours to train. For the DPO experiments, we carry out a grid search using hyperparameters learning rate as 1e-5, and 3e-5, β of 0.1, and 0.5 and number of epochs as 1 and 2 to arrive at the best-performing hyperparameters as mentioned above.

Methods. As baselines, we perform zero-shot prompting of the LLama 2 Chat model[5] (Touvron et al., 2023), denoted by **LLama**, to generate all possible Socratic questions for the current conversation turn. We also prompt LLama 2 in a chain-of-thought (Wei et al., 2022) manner to first generate the current student misconceptions and then generate the Socratic questions, denoted by **LLama (CoT)**.

To decode our trained (SFT and DPO) LLM, we use two decoding techniques, greedy and nucleus sampling, with a p value of 0.9 temperature of 1, and a number of return sequences of 5. We refer to these methods coupled with the trained SFT method as **SFT Greedy**, **SFT Sample-5**, and similarly for the DPO methods. Greedy decoding takes 30 minutes to complete, whereas Sample-5 takes an hour.

Metrics. To measure the similarity between the generated Socratic questions and the ground truth questions, we use two commonly used evaluation metrics in natural language generation tasks: **BERTScore** (Zhang* et al., 2020) based on the DeBERTa language model (He et al., 2021), which measures the semantic similarity, and **Rouge-L** (Lin, 2004), which measures n-gram overlap based

[2] https://openai.com/

[3] https://huggingface.co/codellama/CodeLLama-7b-hf

[4] https://huggingface.co/docs/peft

[5] https://huggingface.co/meta-llama/Llama-2-7b-chat-hf

Table 1: Performance comparison across different methods. All GPT baseline results are reported in (Al-Hossami et al., 2024). Boldface represents the highest value/s for that column.

Method	Rouge-L			BERTScore		
	P	R	F1	P	R	F1
GPT-3.5	21.0	14.3	17.0	56.0	43.5	**48.9**
GPT-3.5 (CoT)	20.3	9.7	12.0	61.7	35.8	41.6
GPT-4	14.1	23.3	17.6	35.4	62.6	45.2
GPT-4 (CoT)	5.2	26.6	8.1	12.6	**64.8**	19.5
LLama	12.8	18.6	13.2	36.0	48.3	35.9
LLama (CoT)	13.7	15.5	13.2	42.3	49.0	41.0
SFT Greedy	29.7	13.4	17.2	61.8	29.3	36.8
DPO Greedy	**30.6**	13.3	17.1	**65.9**	32.7	40.3
SFT Sample-5	14.1	26.0	17.1	32.1	62.9	41.1
DPO Sample-5	15.1	**27.9**	**18.3**	34.8	**64.3**	42.0

on the longest common subsequence (LCS). In addition, the dataset we use (Al-Hossami et al., 2024, 2023) provides multiple ground truth Socratic questions at each dialogue turn. To measure the similarity between a set of m LLM-generated questions with a set of n ground truth questions, we adopt the process used in (Al-Hossami et al., 2024), which uses Edmond Blossom algorithm (Galil, 1986) to find the maximum matching in 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 metrics mentioned above. This step guarantees that every ground-truth question corresponds to, at most, one LLM-generated question, inhibiting semantically equivalent LLM generations from artificially inflating the metric scores. The number of True Positives (TP) is the total sum of the weights of all edges in the optimal matching. False Positives (FP) are calculated by summing the difference between every weight of an edge in the matching with 1. Any unmatched LLM-generated question counts 1 towards False Positive. Similarly, any unmatched ground truth question counts 1 towards False Negative (FN). The TP, FP, and FN values are used to compute the precision, recall, and F1 score for a particular metric. The metric penalizes over-generated LLM questions that do not match with any ground truth questions by classifying them as an FP, thus decreasing the precision.

6 Results and Discussions

In the consistency checking step of the data augmentation phase, we see that 72% of the generated questions are considered for the preference dataset creation as 27% of the generated questions are classified as “good” and 1% as “incorrect”. This result shows that GPT-4 is more prone to generate “good” questions for particular dialogue turns than incorrect questions that do not relate to the problem and the dialogue.

For the task of Socratic question generation, Table 1 shows the comparison between different methods on the metrics defined for our task. All the GPT-3.5 and GPT-4 results are taken from prior work (Al-Hossami et al., 2024). We observe that GPT-4 (CoT) has the highest recall and yet the lowest F1 score. This observation is because, GPT-4 generates a large number of Socratic questions a few of which are similar to the ground truth questions, however, a significant fraction of the generated questions do not correspond to any ground truth questions, hence being labeled as false positive, thus decreasing the precision. (Al-Hossami et al., 2024) also carry out manual analysis to show that GPT (CoT) outputs are the best despite having low F1 scores. This observation can be attributed to the fact that GPT (CoT) has the highest recall among all other GPT methods and hence better corresponds to the ground truth questions.

For the baseline methods that use zero-shot LLama prompting, we observe that LLama (CoT) is the best, which shows that chain-of-thought prompting to first generate the students’ current misconceptions followed by the Socratic questions is effective. Among the preference optimization experiments, we see that DPO consistently outperforms SFT. We also observe that the LLama (CoT) performs as well as DPO Greedy in terms of BERTScore F1 as LLama (CoT) generates a higher number of Socratic questions whereas the DPO Greedy method just generates one. Hence, the recall of the DPO Greedy method is lower than that of LLama (CoT). Among decoding variants, we see that the Sample-5 method is better than the Greedy method highlighting the importance of sampling multiple possible Socratic questions instead of just one.

Overall, we see that our preference-optimized models with DPO give the best Rouge-L scores for all precision, recall, and F1 scores with DPO Greedy having the highest precision and DPO Sample-5 having the highest recall and F1 score among all the methods. DPO Greedy has the highest BERTScore precision, whereas DPO Sample-5 has a recall comparable to the best GPT method, GPT-4 (CoT). These results suggest that the DPO-optimized LLama 2-7B model is better than (or as effective as) much larger models like GPT-4 (25 times larger) for Socratic question generation.

Table 2: An example of invalid Socratic questions generated from GPT-4 for a given conversation, which we use to augment the dataset.

Problem	Write a function “top_k(lst: List[int], k: int) -> List[int]” that returns the top k largest elements in the list. You can assume that k is always smaller than the length of the list. Example Case: top_k([1, 2, 3, 4, 5], 3) => [5, 4, 3]; top_k([-1, -2, -3, -4, -5], 3) => [-1, -2, -3]
Bug Description	The function removes the element at index ‘max(lst)’ instead of removing an element equal to ‘max(lst)’. Consequently, the function throws an IndexError on line 5 when a removed value in ‘lst’ is greater than the length of ‘lst’.
Bug Fixes	On line 5, replace ‘lst.pop(max(lst))’ with ‘lst.remove(max(lst))’
Conversation	**Student**: Hi. I am confused. My code doesn’t seem to work. Can you help? **Instructor**: Hello. Sure, let’s see. Do you know what might be the issue? **Student**: I think the problem is with the ‘.pop()’ method. It seems to have issues with indexing.
Ground Truth	1. Ok, no worries. Let’s review your code line by line. Could you please explain it to me? 2. Let’s start with a simple example. What is the output of the following code snippet: ‘top_k([1, 2, 3, 4, 5], 3)’? 3. Could you please explain what line 5 in your code does? 4. Let’s look into the Python documentation. Can you describe what the ‘.pop()’ method does?
Invalid Generated Questions	**Irrelevant**: What happens if you enter an empty list as the input? **Repeated**: Do you know what might be the issue? **Direct**: Are you sure you should be using the pop() method to remove the maximum element from the list? **Premature**: Have you considered using the remove() method instead of pop()?

7 Case Study

We now use a case study to illustrate why our method leads to better Socratic question generation. First, we show an example of invalid Socratic questions generated by our data augmentation phase. Second, we compare different methods for Socratic question generation.

Table 2 shows an example of the augmented data, i.e., invalid questions generated by GPT-4 for an example problem, which asks students to write code to return the largest k elements in a list. The student’s code (Table 4 Code 1) incorrectly removes elements at index max(lst) as opposed to removing elements equal to max(lst), thereby causing an IndexError. The potential fix to the code is to replace the .pop() function with .remove(). In the conversation, we see that the student knows the problem lies in their use of .pop(). The ground truth Socratic questions for this dialogue turn are highly generic, asking the student to review the code line by line, apply an example test case, or do further reading on Python documentation. We see that the four types of invalid questions generated by GPT-4 are: the *irrelevant* question is out of context and does not help the student understand the bug in their code. The *repeated* question has already been mentioned by the instructor. The *direct* questions reveal the problematic function .pop() and do not give room for the students to discover the problem themselves. The *premature* question directly suggests a code change to replace the .pop() with .remove() function even before the student has realized the actual bug. These diverse examples of invalid questions serve as good training data to let an LLM know what kinds of invalid questions it should avoid generating.

Table 3 shows the questions generated by different methods for an example problem that asks students to write code that returns the factorial of a number. The student’s code (Table 4 Code 2) has some indexing errors with the range function, resulting in the fact variable being multiplied by 0 when the loop starts, hence resulting in the output of 0 no matter what the input is. In the conversation, we see that the ground truth questions encourage the student to debug the code by printing the value of lines 5 and 6 to examine the variables fact and i along with the role of range(n). The LLama output is very verbose and directly reveals that the problem is in the updation of the ‘fact’ variable. The LLama (CoT) output is very vague as it does not refer to the exact line of the code. The SFT Greedy output correctly asks the student the value of i but does not provide more details, such as the value of n. The DPO Greedy method is more helpful since it asks the student to check the value of fact specifically for n=1. The first and fifth questions by SFT Sample-5 are invalid and directly ask the value of fact when i is 0, which directly reveals the problem that fact is always 0. The third

Table 3: Socratic questions generated by different methods.

Problem	Write a function 'factorial(n:int) -> int' that computes the factorial n! of a natural number n, which is defined mathematically as: $0! = 1$ and $n! = n \times (n-1)!$. Additionally, if the input integer n is negative the function should return 0. Example Cases: factorial(-1) => 0; factorial(0) => 1; factorial(1) => 1; factorial(5) => 120
Bug Description and Fix	On line 6, 'fact' is first multiplied with 0 in the first iteration of the for loop and hence remains 0 throughout the loop. To fix the bug, replace 'i' with '(i + 1)' in line 6 and replace 'range(n)' with 'range(1, n + 1)' in line 5.
Conversation	**Student**: Hi! I implemented the factorial function but it doesn't work and I do not know why. Can you help? **Instructor**: Sure. Can you tell me for what values of n it fails and what values it returns in those cases? **Student**: For n = 1 or larger it returns the same value, 0.
Ground Truth	1. Let's see what happens when 'n' is 1. What is the first value that is assigned to variable 'i' in line 5? 2. 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'? 3. 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'?
LLama	So, it seems that the issue is that the value of 'fact' is not being updated correctly in the loop. Can you tell me why do you think 'fact' is not being updated?
LLama (CoT)	Can you explain why you think the bug is in that specific line?
SFT Greedy	What is the value of 'i' in the first iteration of the for loop?
DPO Greedy	Okay. Let's start with 'n = 1'. What do you think 'fact' is on line 6?
SFT Sample-5	1. What is the value of 'fact' at line 6 when 'i' is 0? 2. Let's take a closer look at the line where 'fact' is multiplied. Could you explain the steps you take to compute that multiplication? 3. 0 is not a factorial, right? What do you think the first step should be? 4. What does this tell us about the implementation? 5. Are you multiplying the initial 'fact' value of 1 by 'i' at each step?
DPO Sample-5	1. Alright, so for any value of 'n' that we pass to the function, what is the output of line 6? 2. Can you tell me what value of 'i' is returned in line 6 when 'n = 1'? 3. Okay. Let's look at 'factorial(1)' and 'factorial(2)'. What are the values of 'i' and 'fact' when it goes into the for loop in those cases? 4. Okay, let's go line by line. Let's start from line 5. What do we see? 5. Okay. Can you tell me the steps for computing 'factorial(n)' where 'n' equals 1?

Table 4: Buggy codes corresponding to the case studies

Code 1:

```
def top_k(lst, k):
 result = []
 for i in range(k):
  result.append(max(lst))
  lst.pop(max(lst))
 return result
```

Code 2:

```
def factorial(n):
 if n < 0:
  return 0
 fact = 1
 for i in range(n):
  fact = fact * i
 return fact
```

and fourth outputs are either irrelevant or repeated. The second question, which asks the student to examine the value of `fact` is valid since it does not directly reveal the bug. In contrast, most of the DPO Sample-5 questions are valid, since they urge the student to examine the value of `i` and `fact` on lines 5 and 6 with specific values of `n`, without directly revealing the bug that `i` is always 0. Through these comparisons, we see that DPO improves Socratic question generation compared to SFT and that DPO Sample-5 is highly capable of generating valid yet diverse questions.

8 Conclusions and Future Work

In this work, we propose a method for Socratic question generation in programming problem feedback scenarios. Our method consists of a data augmentation phase to create a preference dataset by synthetically generating invalid questions according to four possible categories. We then use this preference dataset to fine-tune an open-source LLM, LLama 2-7B, using direct preference optimization (DPO). Our results show that the preference-optimized LLama 2-7B model often outperforms existing state-of-the-art prompting methods (on common text similarity metrics) that rely on much larger GPT models (25 times larger), by avoiding invalid questions after training on the augmented dataset. Our method paves the way toward an open-source, accessible, cheaper, privacy-preserving, yet effective alternative to generating Socratic questions which can improve students' learning outcomes without having to rely on proprietary rate-based API-accessed models like GPT-4. There are several avenues for future work. First, we can develop a technique to differentiate types of invalid Socratic questions and not treat them equally while performing preference optimization. This technique would require us to modify the inherent objective function of DPO to incorporate more than one unpreferred question for a single preferred question, which may give us fine-grained control over the LLM generations. Second, we can experiment with open-source LLMs that are larger than 7B to see whether DPO provides more significant gains over SFT on larger models on the Socratic question generation task. Third, we can perform a systematic human evaluation to compare the performance of our proposed method with other baselines. Also, we can focus on designing an automatic metric (based on LLM prompting (Liu et al., 2023)) other than Rouge and BERTScore which captures the helpfulness of the Socratic questions without heavily relying on assigning higher scores only to questions that have high lexical overlap with the ground-truth questions. Fourth, we can experiment with alternative preference optimization methods, such as KTO (Ethayarajh et al., 2024) which do not need explicit preference data in the form of pairs of valid and invalid questions. Fifth, we can also explore if Socratic question generation helps in improving other tasks in computer science education like test case generation (Kumar and Lan, 2024) by posing the problem as answering several Socratic sub-questions (Shridhar et al., 2022). Finally, we can also explore how to make Socratic question generation knowledge-aware, i.e., generating different questions for students with different knowledge states, which can be estimated using the open-ended knowledge tracing method for computer science education (Liu et al., 2022).

9 Limitations

Our work proposes a method for preference optimizing open-source LLMs like LLama 2 for the task of Socratic question generation for student code debugging. We use only LLama 2 as the base model for carrying out preference optimization, and not other open-source models like Mistral (Jiang et al., 2023). Since our main contribution is the data augmentation and preference optimization method, we use only one of the best models open-source models (LLama 2) to show that our method outperforms state-of-the-art models like GPT-4. Future work can also explore the performance of different open-source models using a variety of optimization methods including our data augmentation and preference optimization method for Socratic question generation. Also, we do not formally analyze any biases that exist in the generated augmenting data or the generated Socratic questions. Future work can focus on measuring such biases to make our methods that use these LLMs more inclusive for all students belonging to different demographics.

10 Ethics Policy

Since our invalid questions are generated using an LLM potential linguistic or cultural bias related to the pre-training of the LLM might be reflected. However, we hypothesize that this bias would be minimal as Socratic questions are goal-driven, concise, and framed in the second-person perspective directed toward the student. Our work focuses on open-source LLMs like LLama for Socratic question generation as compared to rate-based API-accessed models like GPT-4 (which is used only once during data augmentation) which implies that our methods are privacy-preserving and there is minimal chance of leakage of students' confidential data. However, training LLMs like LLama on GPUs like A100 for 10 hours results in the emission of CO2 which might not be environmentally friendly.

11 Acknowledgments

We thank Hunter McNichols, Jaewook Lee, Alexander Scarlatos, and Nigel Fernandez for their helpful discussions around this work. The authors are partially supported by the NSF under grant DUE-2215193.

References

Erfan Al-Hossami, Razvan Bunescu, Justin Smith, and Ryan Teehan. 2024. Can language models employ the socratic method? experiments with code debugging. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, SIGCSE 2024, page 53–59, New York, NY, USA. Association for Computing Machinery.

Erfan Al-Hossami, Razvan Bunescu, Ryan Teehan, Laurel Powell, Khyati Mahajan, and Mohsen Dorodchi. 2023. Socratic questioning of novice debuggers: A benchmark dataset and preliminary evaluations. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 709–726, Toronto, Canada. Association for Computational Linguistics.

Nischal Ashok Kumar, Nigel Fernandez, Zichao Wang, and Andrew Lan. 2023. Improving reading comprehension question generation with data augmentation and overgenerate-and-rank. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 247–259, Toronto, Canada. Association for Computational Linguistics.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

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, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Yuhao Dan, Zhikai Lei, Yiyang Gu, Yong Li, Jianghao Yin, Jiaju Lin, Linhao Ye, Zhiyan Tie, Yougen Zhou, Yilei Wang, et al. 2023. Educhat: A large-scale language model-based chatbot system for intelligent education. *arXiv preprint arXiv:2308.02773*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

Zvi Galil. 1986. Efficient algorithms for finding maximum matching in graphs. *ACM Computing Surveys (CSUR)*, 18(1):23–38.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Yann Hicke, Anmol Agarwal, Qianou Ma, and Paul Denny. 2023. Chata: Towards an intelligent question-answer teaching assistant using open-source llms. *arXiv preprint arXiv:2311.02775*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

James M Joyce. 2011. Kullback-leibler divergence. In *International encyclopedia of statistical science*, pages 720–722. Springer.

Majeed Kazemitabaar, Runlong Ye, Xiaoning Wang, Austin Z Henley, Paul Denny, Michelle Craig, and Tovi Grossman. 2024. Codeaid: Evaluating a classroom deployment of an llm-based programming assistant that balances student and educator needs. *arXiv preprint arXiv:2401.11314*.

Julia Kreutzer, Joshua Uyheng, and Stefan Riezler. 2018. Reliability and learnability of human bandit feedback for sequence-to-sequence reinforcement learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788, Melbourne, Australia. Association for Computational Linguistics.

Nischal Ashok Kumar and Andrew Lan. 2024. Using large language models for student-code guided test case generation in computer science education. *arXiv preprint arXiv:2402.07081*.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Naiming Liu, Zichao Wang, Richard Baraniuk, and Andrew Lan. 2022. Open-ended knowledge tracing for computer science education. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3849–3862.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

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.

Richard Paul and Linda Elder. 2007. Critical thinking: The art of socratic questioning. *Journal of developmental education*, 31(1):36.

Chris Quintana, Brian J Reiser, Elizabeth A Davis, Joseph Krajcik, Eric Fretz, Ravit Golan Duncan, Eleni Kyza, Daniel Edelson, and Elliot Soloway. 2018. A scaffolding design framework for software to support science inquiry. In *Scaffolding*, pages 337–386. Psychology Press.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh*

Conference on Neural Information Processing Systems.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

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*, pages 27–43.

Alexander Scarlatos, Digory Smith, Simon Woodhead, and Andrew Lan. 2024. Improving the validity of automatically generated feedback via reinforcement learning. *arXiv preprint arXiv:2403.01304*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

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.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

Henansh Tanwar, Kunal Shrivastva, Rahul Singh, and Dhruv Kumar. 2024. Opinebot: Class feedback reimagined using a conversational llm. *arXiv preprint arXiv:2401.15589*.

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*.

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, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256.

David Wood, Jerome S Bruner, and Gail Ross. 1976. The role of tutoring in problem solving. *Journal of child psychology and psychiatry*, 17(2):89–100.

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*.

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*.

Scoring with Confidence? –
Exploring High-confidence Scoring for Saving Manual Grading Effort

Marie Bexte[1] **Andrea Horbach**[1, 2] **Lena Schützler**[3]
Oliver Christ[3] **Torsten Zesch**[1]
[1]CATALPA, FernUniversität in Hagen, Germany
[2]Hildesheim University, Germany
[3]FernUniversität in Hagen, Germany

Abstract

A possible way to save manual grading effort in short answer scoring is to automatically score answers for which the classifier is highly confident. We explore the feasibility of this approach in a high-stakes exam setting, evaluating three different similarity-based scoring methods, where the similarity score is a direct proxy for model confidence. The decision on an appropriate level of confidence should ideally be made before scoring a new prompt. We thus probe to what extent confidence thresholds are consistent across different datasets and prompts. We find that high-confidence thresholds vary on a prompt-to-prompt basis, and that the overall potential of increased performance at a reasonable cost of additional manual effort is limited.

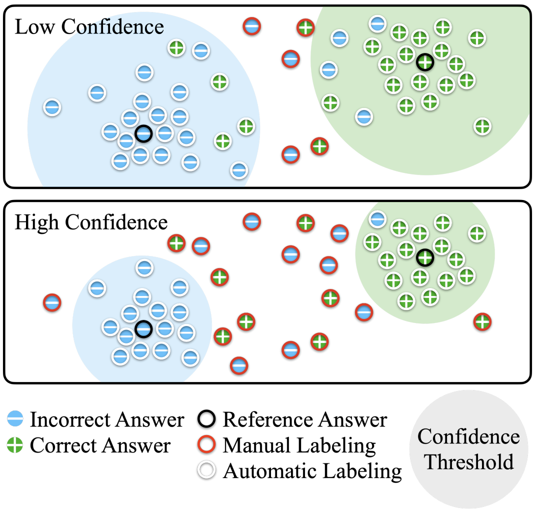


Figure 1: Confidence-based scoring

1 Introduction

Whenever a (semi-)automatic process is used to assist humans in scoring free-text answers, there is a trade-off between the human workload required and the resulting scoring accuracy. Without any human input, the accuracy of the automated rating is usually quite low (Egaña et al., 2023), however, already little human input might go a long way in improving the automation quality. Suen et al. (2023) score answers in a setting that uses reference answers and operationalize the confidence of the model as the similarity to the closest reference answer. This concept is visualized in Figure 1. They find that setting a threshold on model confidence, deferring to manual evaluation what falls short of it, leads to reasonable manual effort and high scoring accuracy.

We test the applicability of this method in a high-stakes classroom setting, where items are usually not re-used. This sharply limits the amount of manual scoring effort that can be spent before automation becomes uneconomical. We thus use a small volume of reference answers and examine to what extent a sensible pre-set confidence threshold can be established. As we cannot make the high-stakes student answers publicly available, we additionally replicate our results on four widely used datasets.

Our study makes the important step of linking state-of-the-art natural language processing for rating free-text items with the practical questions of start-up costs for building the models.

2 Related Work

The idea to automatically score only parts of all answers or to defer answers with a particularly low confidence of the algorithm to human scoring has been explored before (Funayama et al., 2020, 2022). The approach that is closest to ours is Suen et al. (2023), where answers to medical exam questions are scored using a similarity-based scoring method (Bexte et al., 2022, 2023) and the confidence of the classifier is operationalized through the similarity to the closest reference answer. This method could be taken further to iteratively improve a classifier through those human-labeled low-confidence answers, i.e. using Active Learning (Settles, 2009), as in the scoring domain done by Horbach and Palmer

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 119–124
June 20, 2024 ©2024 Association for Computational Linguistics

			Answer averages across prompts			
Dataset	**# Prompts**	**# Labels**	**#**	**% Unique**	**Length**	**Language**
UniversityExams	7	2	544	34	18	German
ASAP	10	3 or 4	2,227	100	239	English
Beetle	56	2	93	100	49	English
SEB	140	2	42	97	64	English
Powergrading	10	2	678	35	25	English

Table 1: Answer and label statistics of the datasets used in our experiments.

(2016) and Kishaan et al. (2020). Such a procedure does however have the disadvantage that human annotators have to annotate small batches of answers over a longer period of time.

Other studies rely on the idea that similar answers should receive the same score. Such a grouping of answers could be reached through surface-level normalization (cf. Zehner et al. (2016)), which reduces orthographic variance, or unsupervised clustering methods operating on the surface level (Horbach et al., 2014; Zesch et al., 2015; Horbach and Pinkal, 2018; Weegar and Idestam-Almquist, 2023), on the semantic level using, e.g. LSA approaches (Zehner et al., 2016; Andersen et al., 2023), or a combination of the two (Basu et al., 2013).

3 Data

We conduct experiments on five datasets (see Table 1). Our high-stakes exam dataset consists of German answers collected from university students as part of their final exam in a statistics class. We refer to this dataset as UniversityExams. It contains 7 prompts that each require a short answer. An exemplary question (translated from German) is *Name the method that is used to estimate the required sample size before an experiment*, where a satisfactory answer would be *a-priori power analysis*. Answers are labeled on a binary scale as either correct or incorrect. Due to the sensitiveness of this data, we can unfortunately not publish it.

We thus also run experiments on four existing, publicly available English datasets, that we use to put results on the exam data into context: The **ASAP**[1] dataset consists of answers to ten prompts from the domains of Biology, Science, and English Language Arts. **Powergrading** (Basu et al., 2013) has answers to ten United States Citizenship Exam questions that were collected from Amazon Mechanical Turk. The Student Response Analysis (SRA) dataset (Dzikovska et al., 2013) is split into two subsets: **Beetle** and **SciEntsBank (SEB)**. Beetle has answers to 56 questions about electricity and electronics, while SciEntsBank contains answers to 150[2] prompts that are from a mix of 15 different science domains. We use the two-way labeled version of the SRA dataset, where answers are classified as correct or incorrect.

4 Experimental Setup

Data Split We split the answers to each prompt into reference and test answers. Our reference answers aim to simulate a teacher manually providing exemplary answers for the different outcome labels. In practice, this would mean a rather small volume of unique examples per label. For each prompt, we thus randomly sample 5 answers per label as references, ensuring that there are no duplicates in this sample. Whenever a similarity metric is fine-tuned on the reference answers, we split them into four answers per label to train and one answer per label to validate.

Classifiers We compare three methods of similarity-based classification that differ with respect to the employed similarity metric. All use a set of reference answers to label the test answers: Based on the respective similarity metric, we predict the label of the most similar reference answer. We compare the following metrics: (i) Edit distance[3] and two variants of cosine similarity based on (ii) pretrained or (iii) fine-tuned SBERT embeddings (Reimers and Gurevych, 2019).[4] For the English datasets, we use the *all-MiniLM-L6-v2* base model, and for the German data the *paraphrase-multilingual-MiniLM-L12-v2* one, both taken from HuggingFace.

[1] https://www.kaggle.com/c/asap-sas/overview

[2] We combine answers from *training* and *unseen questions*. Since our experiments require at least five answers for each label, we can only use 140 prompts.

[3] Determined using the python *Levenshtein* module: https://github.com/rapidfuzz/Levenshtein

[4] We transform edit distance into a noralized similarity for better comparability by computing *1-edit distance* and scaling by length of the longest answer to the respective prompt.

Dataset	Edit	SBERT		target
		pretrained	finetuned	
UniversityExams	.86	.86	.91	.95
ASAP	.46	.43	.50	.60
Beetle	.65	.65	.68	.80
SEB	.68	.65	.71	.80
Powergrading	.87	.92	.93	1.00

Table 2: Weighted F1 results when all test answers are scored fully automated.

To fine-tune SBERT, we follow the approach by Bexte et al. (2022): We train with pairs of answers that are labeled with a similarity label of 1 if both answers have the same score and 0 otherwise. To form these training examples, we pair each training answer with each other training answer. To validate, we pair each validation answer with each training answer. At inference, each test answer is compared to each training and each validation answer, i.e. all reference answers. We train for 30 epochs with a batch size of 8, using an *OnlineContrastiveLoss* and an *EmbeddingSimilarityEvaluator*.

Evaluation We evaluate using weighted F1, reporting averages across all prompts of a dataset.

5 Experiments

First, we report results of a **fully automatic baseline**. In this approach, all test answers are scored automatically, i.e. assigned the label of the most similar reference answer. We then explore **confidence-based scoring**, only scoring instances where similarity exceeds a given threshold automatically. The remaining answers are referred to a human for manual scoring. The fully automatic baseline can be seen as an extreme case of this threshold-based scoring, where the confidence threshold is set so that all classifier decisions are accepted. We speak of a baseline, as introducing a confidence threshold should discard misclassifications and thus increase scoring performance.

5.1 Fully-automated Baseline

Table 2 shows performance of our three scoring methods on the fully-automated baseline, i.e. when all test answers are labeled automatically. It is apparent that some datasets are easier to score than others, with a rather consistent pattern across scoring methods. Particularly the UniversityExams and Powergrading answers are easier to score, which is in part due to the lower percentage of unique answers in these datasets. Overall, there is a slight advantage of the fine-tuned SBERT over the other methods.

5.2 Confidence-based Scoring

Using a similarity-based approach to score answers brings about the benefit of being able to take the similarity on which the classification hinges as a confidence estimate. Suen et al. (2023) were able to increase performance by deferring answers where the model is not confident enough to manual labeling. This requires a predefined threshold that dictates whether to take the predicted label or seek manual labeling. In a practical setting, there should not be a requirement of having to determine this threshold for each new prompt, as this would require substantial amounts of labeled data for the new prompt, thereby diminishing the advantage of automatic evaluation. To assess whether there is such a threshold that is reasonable to assume for new prompts, we analyze how much well-suited thresholds vary between datasets and prompts.

Data-driven Threshold Selection To decide on a suitable threshold for each prompt, we define a target performance for each dataset. These values are listed in Table 2 (under column 'target') and were chosen to push performance around .10 weighted F1 above the fully-automated baseline. Figure 4 in the Appendix shows that performance of the individual prompts in a dataset varies: For some prompts, the target performance was already reached (or surpassed), while others lie beneath it, at times substantially. For these, we calculate the lowest possible threshold value that reaches the target performance[5]. Weighted F1 is then calculated on all answers for which the model's confidence exceeds this threshold, i.e. calculated only on those answers for which the machine-predicted label is taken. Answers that are deferred to manual labeling are excluded from the performance calculation, as they are by definition assumed to be scored correctly.

Figure 2 (blue bars) shows the determined optimal thresholds, with each bar corresponding to a prompt of the respective dataset. The only dataset where thresholds are somewhat close together is edit distance-based scoring of Powergrading, where they range from .92 to .99. Otherwise, thresholds vary widely, indicating that it is difficult to predefine a threshold to apply to a new prompt. On top

[5] Prompts already at target level have a threshold of 0.

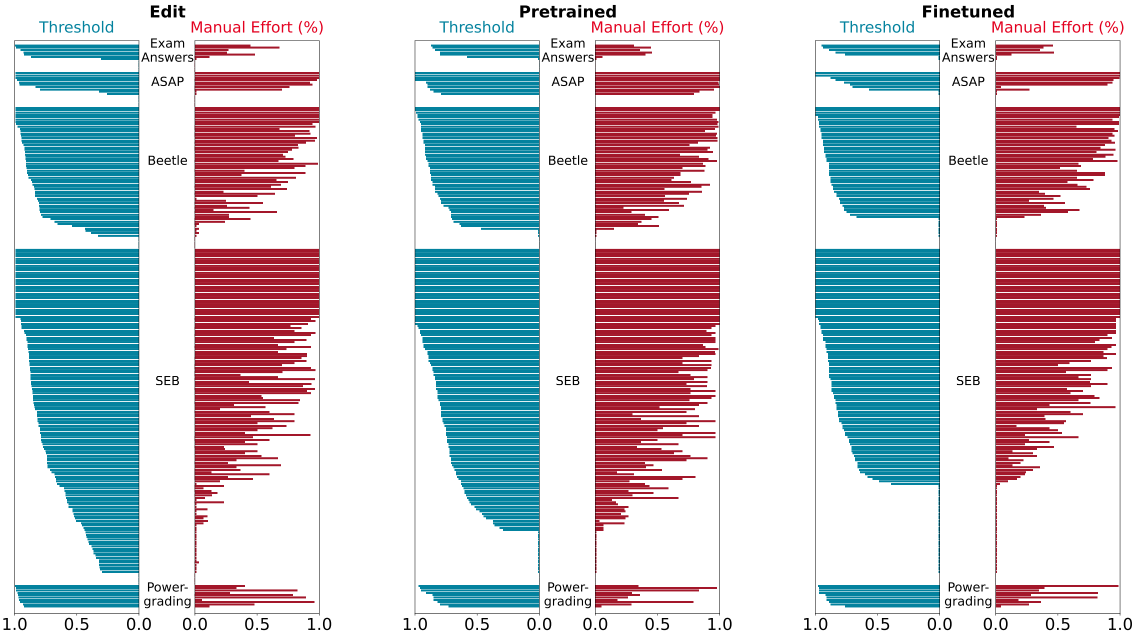


Figure 2: Prompt-wise depiction of thresholds that would have to be set in order to achieve the target performance level (see Table 2). Red bars indicate how much test data falls below the threshold, i.e. has to be scored manually.

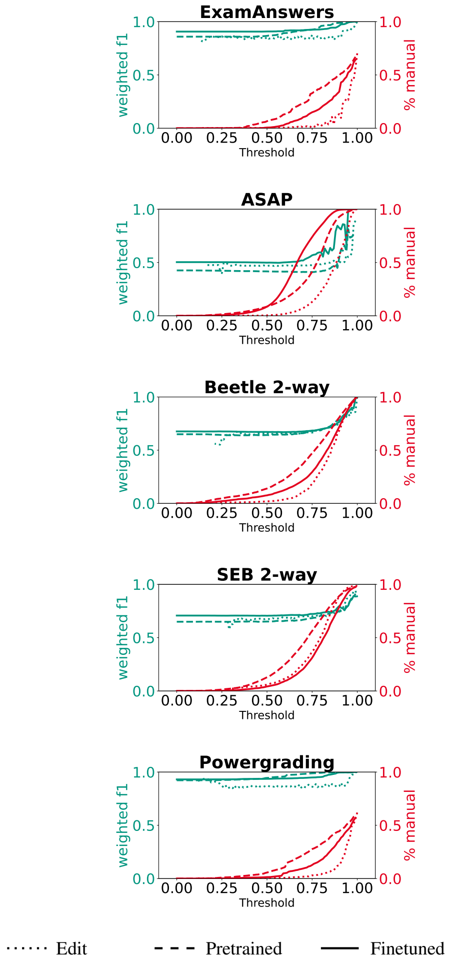


Figure 3: Weighted F1 and amount of answers that requires manual scoring averaged over all prompts of the respective dataset.

of the rather wide span of optimal thresholds, the red bars depict how much of the test data would be deferred to manual labeling. We see that for many of the threshold values, this would make up a substantial amount of answers, often over half of them. Thus, even if there is a threshold found, reaching the target performance level comes at the cost of a large volume of manual annotation effort.

Predefining Threshold Values Instead of a data-driven search for an optimal threshold value, one could also make a top-down decision on a reasonably seeming threshold. Our next analysis inspects how threshold values are related to performance and manual correction effort. Figure 3 shows the relation between threshold value, performance and manual effort averaged over all prompts of a dataset. In general, performance tends to be stable for a rather wide range of thresholds, and only starts to increase when substantial manual effort is required. There is thus no general potential of increasing performance at a reasonable cost of additional manual labeling.

6 Conclusion

While previous work showed that confidence-based scoring can be successful (Suen et al., 2023), we do not find this to hold in our experiments. This may in part be due to the lower volume of reference answers and the higher overall scoring difficulty of some of the datasets we use. On some prompts, there may be thresholds that lead to a desirable tradeoff between manual effort and performance increase, but we did not find a general range of threshold values that would be promising to apply to unseen prompts.

Limitations

Due to the sensitive nature of the exam data, we can unfortunately not publish it. This limits the reproducibility of our results.

When we set thresholds on the similarity, we calculate performance based on only those examples that exceed the confidence threshold. One could also argue to include the answers that are deferred to manual labeling as correctly classified examples. This would increase performance, but it would also mean that a certain volume of answers might be scored with substantially inferior performance, as it would enable for manually labeled answers to even out misclassifications by a model. In practice we want to guarantee a certain level of performance for all students, and hence calculate performance solely on those answers that are classified by a model.

Ethical Considerations

The motivation for this work was to assess the usefulness of automated confidence-based scoring in a high-stakes setting. The performance levels on the SRA and ASAP datasets are however a long way off from being reliable enough for employment in an actual classroom. Even on the better-performing Powergrading and UniversityExams data, the local legal situation is likely to put significant conditions on the use of automated decisions, or even prohibit this entirely.

Achnowledgements

This work was partially conducted at “CATALPA - Center of Advanced Technology for Assisted Learning and Predictive Analytics” of the FernUniversität in Hagen, Germany.

References

Nico Andersen, Fabian Zehner, and Frank Goldhammer. 2023. Semi-automatic coding of open-ended text responses in large-scale assessments. *Journal of Computer Assisted Learning*, 39(3):841–854.

Sumit Basu, Chuck Jacobs, and Lucy Vanderwende. 2013. Powergrading: a clustering approach to amplify human effort for short answer grading. *Transactions of the Association for Computational Linguistics*, 1:391–402.

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, Seattle, Washington. Association for Computational Linguistics.

Marie Bexte, Andrea Horbach, and Torsten Zesch. 2023. Similarity-based content scoring-a more classroom-suitable alternative to instance-based scoring? In *Findings of the association for computational linguistics: Acl 2023*, pages 1892–1903.

Myroslava Dzikovska, Rodney Nielsen, Chris Brew, Claudia Leacock, Danilo Giampiccolo, Luisa Bentivogli, Peter Clark, Ido Dagan, and Hoa Trang Dang. 2013. SemEval-2013 Task 7: The Joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 263–274, Atlanta, Georgia, USA. Association for Computational Linguistics.

Aner Egaña, Itziar Aldabe, and Oier Lopez de Lacalle. 2023. Exploration of annotation strategies for automatic short answer grading. In *Artificial Intelligence in Education*, pages 377–388, Cham. Springer Nature Switzerland.

Hiroaki Funayama, Shota Sasaki, Yuichiroh Matsubayashi, Tomoya Mizumoto, Jun Suzuki, Masato Mita, and Kentaro Inui. 2020. Preventing critical scoring errors in short answer scoring with confidence estimation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 237–243.

Hiroaki Funayama, Tasuku Sato, Yuichiroh Matsubayashi, Tomoya Mizumoto, Jun Suzuki, and Kentaro Inui. 2022. Balancing cost and quality: an exploration of human-in-the-loop frameworks for automated short answer scoring. In *International Conference on Artificial Intelligence in Education*, pages 465–476. Springer.

Andrea Horbach and Alexis Palmer. 2016. Investigating active learning for short-answer scoring. In *Proceedings of the 11th workshop on innovative use of NLP for building educational applications*, pages 301–311.

Andrea Horbach, Alexis Palmer, and Magdalena Wolska. 2014. Finding a tradeoff between accuracy and rater's workload in grading clustered short answers. In *LREC*, pages 588–595.

Andrea Horbach and Manfred Pinkal. 2018. Semi-supervised clustering for short answer scoring. In

Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).

Jeeveswaran Kishaan, Mohandass Muthuraja, Deebul Nair, and Paul G Plöger. 2020. Using active learning for assisted short answer grading. In *ICML 2020 Workshop on Real World Experiment Design and Active Learning*.

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.

Burr Settles. 2009. Active learning literature survey.

King Yiu Suen, Victoria Yaneva, Le An Ha, Janet Mee, Yiyun Zhou, and Polina Harik. 2023. ACTA: Short-answer grading in high-stakes medical exams. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 443–447, Toronto, Canada. Association for Computational Linguistics.

Rebecka Weegar and Peter Idestam-Almquist. 2023. Reducing workload in short answer grading using machine learning. *International Journal of Artificial Intelligence in Education*, pages 1–27.

Fabian Zehner, Christine Sälzer, and Frank Goldhammer. 2016. Automatic coding of short text responses via clustering in educational assessment. *Educational and psychological measurement*, 76(2):280–303.

Torsten Zesch, Michael Heilman, and Aoife Cahill. 2015. Reducing annotation efforts in supervised short answer scoring. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 124–132.

A Detailed Results of Fully-automated Baseline

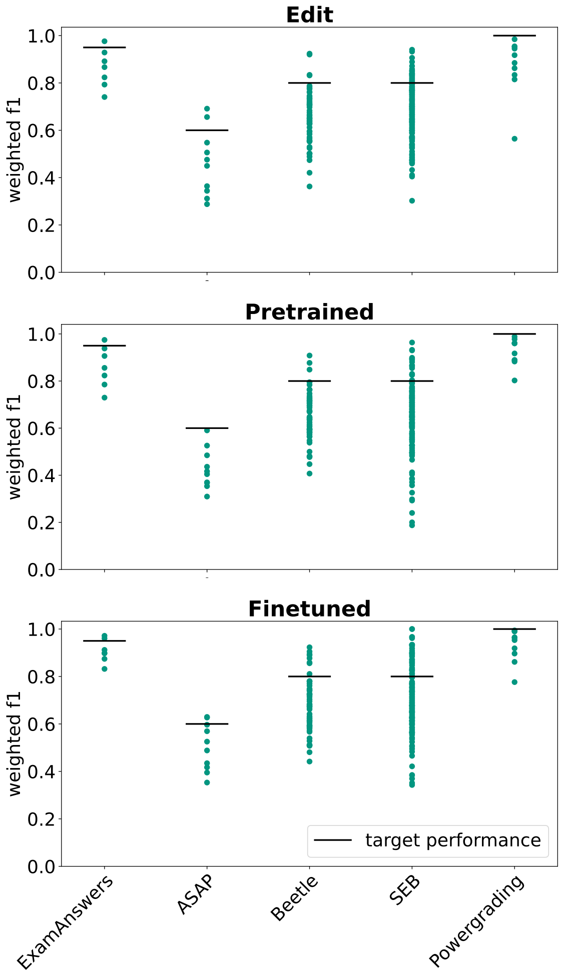


Figure 4: Prompt-wise results for the fully automated baseline and target performance for the respective datasets.

Predicting Initial Essay Quality Scores to Increase the Efficiency of Comparative Judgment Assessments

Michiel De Vrindt [1ac] **Anaïs Tack** [1ad] **Renske Bouwer** [2b]
Wim Van Den Noortgate [1ac] **Marije Lesterhuis** [3e]

[1] KU Leuven [a] imec research group itec [b] Institute for Language Sciences
[c] Faculty of Psychology and Educational Sciences [2] Utrecht University [3] UMC Utrecht
[d] Faculty of Arts [e] Center for Research and Development of Health Professions Education

Abstract

Comparative judgment (CJ) is a method that can be used to assess the writing quality of student essays based on repeated pairwise comparisons by multiple assessors. Although the assessment method is known to have high validity and reliability, it can be particularly inefficient, as assessors must make many judgments before the scores become reliable. Prior research has investigated methods to improve the efficiency of CJ, yet these methods introduce additional challenges, notably stemming from the initial lack of information at the start of the assessment, which is known as a cold-start problem. This paper reports on a study in which we predict the initial quality scores of essays to establish a warm start for CJ. To achieve this, we construct informative prior distributions for the quality scores based on the predicted initial quality scores. Through simulation studies, we demonstrate that our approach increases the efficiency of CJ: On average, assessors need to make 30% fewer judgments for each essay to reach an overall reliability level of 0.70.

1 Introduction

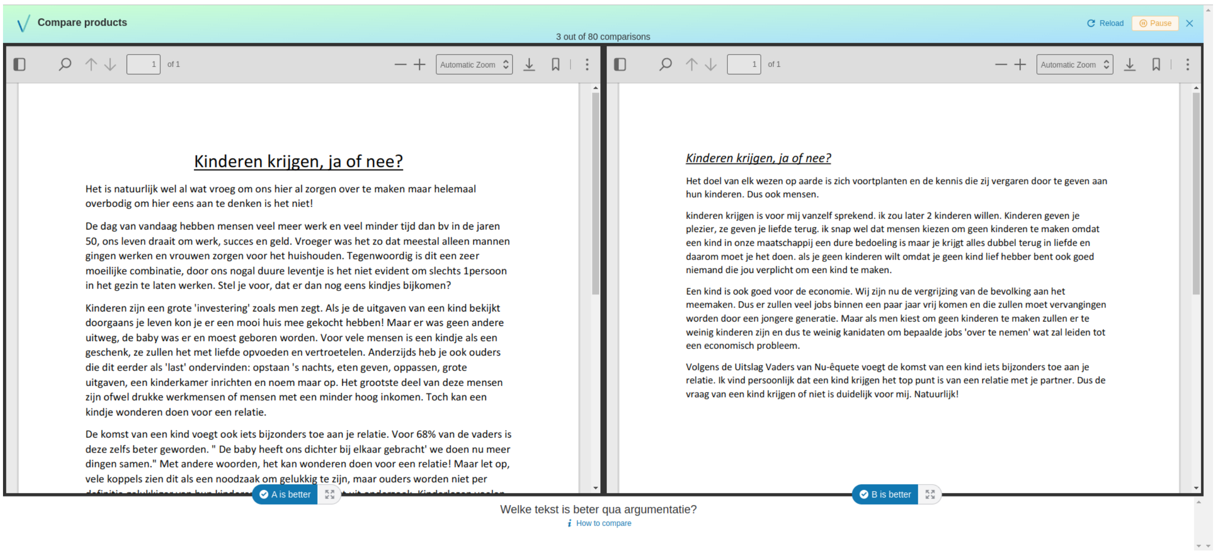
The Comparative Judgment (CJ) method is utilized in diverse educational assessments, and specifically, some educational institutions employ it for the assessment of student essays. As shown in Figure 1, this approach involves presenting two essays in a web-based tool, where assessors compare them to determine the best one. After a sufficient number of judgments, all pairwise comparisons are used to calculate a quality score for each essay. In contrast to rubric marking, CJ provides distinctive advantages. Assessors can apply their expertise and experience flexibly, without strict adherence to rubrics (Bloxham, 2009; Laming, 2003). Additionally, CJ enhances the reliability and validity of scores by incorporating multiple judgments from various assessors (Lesterhuis et al., 2022; Verhavert et al., 2019).

Despite the advantages of CJ, it still requires many judgments from assessors before quality scores become reliable enough, typically requiring between 10 and 14 judgments per essay to achieve a reliability level of 0.70 (Verhavert et al., 2019), rendering the assessment method rather inefficient (McMahon and Jones, 2015). A cause of its inefficiency is that, at the start of the assessment, there is no information about the quality scores, as no judgments have been made yet. In adaptive learning systems, this problem is commonly referred to as cold-start problem (Sun et al., 2022a; Pliakos et al., 2019).

A solution to alleviating this cold-start problem, and subsequently increasing the efficiency of CJ, would be to introduce a 'warm start' in the assessment by automatically predicting initial quality scores for essays. Although the prediction of essay quality has already been extensively explored in automated essay scoring (AES) (see a review by Klebanov and Madnani, 2022), these studies have mostly focused on what could be defined as non-comparative, or absolute (Bouwer et al., 2023), essay scoring, where each essay is scored as a standalone piece without comparison to other essays. To the best of our knowledge, there have been few to no studies that explored the automatic prediction of essay quality scores obtained through CJ assessments.

To address this research gap, we studied the extent to which essay quality scores, resulting from a CJ assessment, can be automatically predicted and used to alleviate the cold start of CJ with the goal of increasing the efficiency of CJ for assessing essay quality. We focused on Dutch essays written for argumentative assignments. Firstly, we conducted a machine learning experiment in which deep learning models were trained on data collected from CJ assessments to predict quality scores of essays. Secondly, we ran simulations where we used the predicted quality scores as initial quality scores to

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 125–136
June 20, 2024 ©2024 Association for Computational Linguistics



Kinderen krijgen, ja of nee?

Het is natuurlijk wel al wat vroeg om ons hier al zorgen over te maken maar helemaal overbodig om hier eens aan te denken is het niet!

De dag van vandaag hebben mensen veel meer werk en veel minder tijd dan bv in de jaren 50, ons leven draait om werk, succes en geld. Vroeger was het zo dat meestal alleen mannen gingen werken en vrouwen zorgen voor het huishouden. Tegenwoordig is dit een zeer moeilijke combinatie, door ons nogal duure leventje is het niet evident om slechts 1persoon in het gezin te laten werken. Stel je voor, dat er dan nog eens kindjes bijkomen?

Kinderen zijn een grote 'investering' zoals men zegt. Als je de uitgaven van een kind bekijkt doorgaans je leven kon je er een mooi huis mee gekocht hebben! Maar er was geen andere uitweg, de baby was er en moest geboren worden. Voor vele mensen is een kindje als een geschenk, ze zullen het met liefde opvoeden en vertroetelen. Anderzijds heb je ook ouders die dit eerder als 'last' ondervinden: opstaan 's nachts, eten geven, oppassen, grote uitgaven, een kinderkamer inrichten en noem maar op. Het grootste deel van deze mensen zijn ofwel drukke werkmensen of mensen met een minder hoog inkomen. Toch kan een kindje wonderen doen voor een relatie.

De komst van een kind voegt ook iets bijzonders toe aan je relatie. Voor 68% van de vaders is deze zelfs beter geworden. " De baby heeft ons dichter bij elkaar gebracht' we doen nu meer dingen samen." Met andere woorden, het kan wonderen doen voor een relatie! Maar let op, vele koppels zien dit als een noodzaak om gelukkig te zijn, maar ouders worden niet per

Kinderen krijgen, ja of nee?

Het doel van elk wezen op aarde is zich voortplanten en de kennis die zij vergaren door te geven aan hun kinderen. Dus ook mensen.

kinderen krijgen is voor mij vanzelf sprekend. ik zou later 2 kinderen willen. Kinderen geven je plezier, ze geven je liefde terug. ik snap wel dat mensen kiezen om geen kinderen te maken omdat een kind in onze maatschappij een dure bedoeling is maar je krijgt alles dubbel terug in liefde en daarom moet je het doen. als je geen kinderen wilt omdat je geen kind lief hebber bent ook goed niemand die jou verplicht om een kind te maken.

Een kind is ook goed voor de economie. Wij zijn nu de vergrijzing van de bevolking aan het meemaken. Dus er zullen veel jobs binnen een paar jaar vrij komen en die zullen moet vervangingen worden door een jongere generatie. Maar als men kiest om geen kinderen te maken zullen er te weinig kinderen zijn en dus te weinig kanidaten om bepaalde jobs 'over te nemen' wat zal leiden tot een economisch probleem.

Volgens de Uitslag Vaders van Nu-êquete voegt de komst van een kind iets bijzonders toe aan je relatie. Ik vind persoonlijk dat een kind krijgen het top punt is van een relatie met je partner. Dus de vraag van een kind krijgen of niet is duidelijk voor mij. Natuurlijk!

Figure 1: Screenshot of the Comproved web application (https://comproved.com), showcasing a comparative judgment assessment. Here, two Dutch essays discussing the topic ‘Having children, yes or no?’ are randomly chosen and presented to an assessor, who determines which essay showed the best argumentation.

alleviate the cold start of CJ. These steps were conducted to answer the following research questions:

1. To what extent can current deep learning models automatically predict essay quality scores that resemble quality scores obtained from CJ assessments?

2. If these predicted scores are used as initial quality scores within CJ, to what extent can we decrease the number of comparative judgments needed to obtain reliable scores?

2 Background

2.1 Comparative Judgment Assessments

Generally, CJ assessments consist of three steps that are repeated. In a first step, a pair of two essays is selected and presented to one of the multiple assessors. In a second step, the assessor is tasked with comparing the two essays and determining which is of a higher quality given the task description of the assignment, that is, the prompt. In a third step, statistical models such as the Bradley-Terry-Luce (BTL) model are used to model the outcomes of all pairwise comparisons on a quality scale (Bradley and Terry, 1952; Luce, 1959).

More formally, BTL model relates $\mathbb{P}(i \succ j)$, that is the probability that essay i is preferred over essay j, to the difference in their estimated quality scores, θ_i and θ_j (see Equation 1), with $i \in \{1, \ldots, n\}$ and $i \neq j$. The smaller the difference, the closer the probability is to 0.50. The outcome of comparing essay i with essay j is denoted by $Z_{ij} \in \{0, 1\}$, where $Z_{ij} = 1$ in case essay i is preferred over essay j, and 0 otherwise. Each quality is a logit value $\theta_i \in \mathbb{R}$ where $\sum_{i=1}^{n} \theta_i = 0$.

$$\mathbb{P}(i \succ j) := \mathbb{P}(Z_{ij} = 1) = \frac{e^{\theta_i - \theta_j}}{1 + e^{\theta_i - \theta_j}} \quad (1)$$

$$Z_{ij} \sim \text{Bernoulli}\left(\mathbb{P}(i \succ j)\right) \quad (2)$$

Different selection rules for CJ (step 1) have been proposed to increase the efficiency of the assessment. These selection rules rely on certain characteristics of essays. Most notably, Pollitt (2012) proposed to select pairs of essays adaptively based on the closest estimated quality scores. The outcomes of these judgments are the most uncertain and, therefore, the most informative for the quality scores in a statistical sense. However, there are two drawbacks to adaptive selection: First, it cannot be used at the start of the assessment, as quality scores are still unknown, and second, during the assessment, adaptive selection can lead to an overly optimistic view of reliability, causing the assessment to end prematurely (Bramley and Vitello, 2019; Crompvoets et al., 2020). Alternatively, pairs of essays can be selected based on the textual information of essays. De Vrindt et al. (2022) proposed to select pairs of essays that are semantically similar during the initial phase of the CJ assessment. They encoded the essay texts as numeric vectors using doc2vec (Le and Mikolov, 2014) and selected the pairs with the highest cosine similarity. However, the efficiency gain they observed was only limited. Therefore, it is of interest to investigate other ways of using textual information of essays to speed up CJ assessments. We focus on the automatic prediction of quality scores based on previously assessed essay texts.

2.2 Automated Essay Scoring

In the field of AES, the automatic prediction of scores has been extensively investigated with as goal to reduce the workload of assessors. This field has experienced significant advances driven by deep learning (Ramesh and Sanampudi, 2022). The proposed deep learning techniques depend on the educational setting in which AES is used. In scenarios where no previously scored essays are available, the prediction relies solely on the essay text itself. This can be achieved, for example, through unsupervised learning (Mim et al., 2019; Wang et al., 2023). In AES research, it is typical to have scored essays on hand. These scored essays help researchers understand the connection between scores and essay content, enabling them to predict essay scores more accurately. This can be achieved through supervised learning (Alikaniotis et al., 2016; Taghipour and Ng, 2016; Yang et al., 2020; Li et al., 2022). For supervised learning, essays that have been scored in the training set can be written for a different assignment than the essays in the test set for which scores are predicted. In such a setting, the prompt for the assignments is often considered to predict the essay scores in addition to the essay texts (Li et al., 2020; Do et al., 2023; Liu et al., 2019).

2.3 Cold-start Problem in Psychometry

The cold-start problem is most commonly termed in the context of recommender systems to denote the difficulty of proposing items to users when the preferences of the users or the characteristics of the items are unknown due to limited user interactions. Using language models, this issue has been addressed by extracting characteristics from item texts (Penha and Hauff, 2020) or by generating user preferences based on the textual description of user historical preferences and items (Wang et al., 2024).

Similarly, in computerized adaptive testing, the cold-start problem persists. These systems select test items so that the difficulty of the item matches the test takers ability, but when responses for items are lacking, inferring item difficulty becomes challenging. Therefore, to calibrate the characteristics of the items, responses for the items need to be collected during a pilot phase. To mitigate the need for extensive piloting, Settles et al. (2020) extracted the linguistic features of test items measuring their difficulty. Alternatively, McCarthy et al. (2021) used pre-trained embeddings of test items to estimate their difficulty and discriminatory power.

The cold-start problem for CJ is similar: quality scores for essays are unknown at the start of an assessment because assessors have not judged them, requiring assessors to make many judgments during the assessment. Analogously to recommender systems and computerized adaptive testing, we address the cold start of CJ by inferring the unknown measures, namely the quality scores, from essay texts.

3 Method

3.1 Data

This study was based on data gathered in a previous study by Lesterhuis et al. (2022). The dataset, described in Table 1, comprised three assignments in which students around the age of 16 wrote argumentative essays in Dutch. The topics for these essays were: (1) having children, (2) organ donation, and (3) stress experienced by students. Students were provided with a prompt detailing the essay topic, the task requirements, and the source texts they were required to integrate in the essay.

Assignment	Essays	Tokens	Tokens/Essay
	N		$M \pm SD$
1. Children	135	42,349	316 (± 93)
2. Organ	136	40,990	304 (± 90)
3. Stress	35	11,286	322 (± 103)

Table 1: Overview of the argumentative writing tasks gathered by Lesterhuis et al. (2022). Tokenization was performed using the Dutch tokenizer from spaCy (Explosion, 2023), which splits the essay texts into meaningful segments.

The essays were assessed by secondary education assessors using a comparative judgment method. Assessors were presented with pairs of randomly selected essays and had to decide which one was better in terms of argumentation, as illustrated in Figure 1. The number of assessors for each assignment and the total of judgments per essay are detailed in Table 2.

Assignment	Judgments/Essay	Assessors
1. Children	18	55
2. Organ	13	52
3. Stress	27	42

Table 2: Overview of the number of comparative judgments made per argumentative writing assignment

To study the predictability of initial essay quality scores and their role in a warm start, it is of course imperative to have quality scores for each essay. For each of the three assignments separately, essay quality scores were derived from the parameters of a Bayesian BTL model with a cold-start condition. These model parameters were estimated based on all comparative judgments within the same assignment. Since these parameters reflect the quality scores estimated at the end of the CJ assessment, we will refer to them as the 'final quality scores' throughout the remainder of this paper. Additional details regarding this cold-start model will be provided in Section 3.5. The distributions of the quality score for each essay within each assignment are shown in Figure 2. Given the large number of comparative judgments per essay (Verhavert et al., 2019) and the diverse panel of assessors responsible for these judgments (van Daal et al., 2016), we can confidently affirm the reliability and validity of these estimated scores.

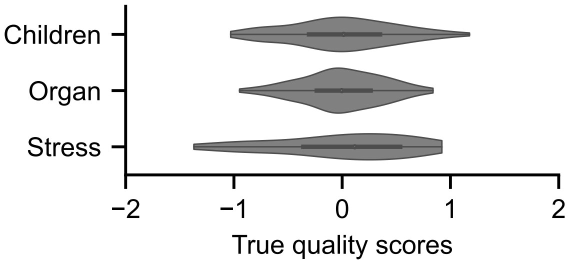


Figure 2: Distributions of final quality scores estimated from a Bayesian BTL model with a cold-start condition

3.2 Models

For predicting essay quality scores, we employed various pre-trained language models and fine-tuned them based on the final quality scores. While alternative feature-based and classical NLP methods exist for this purpose, we focused on fine-tuning transformer models due to their demonstrated superiority in AES research (Uto et al., 2020; Ormerod et al., 2021). We specifically avoided multilingual models, concentrating solely on Dutch models, as prior studies indicate that monolingual models tend to outperform on tasks involving Dutch texts (de Vries et al., 2019; Delobelle et al., 2020). We used three different pre-trained Dutch language models, namely **BERTje** (base, uncased) (de Vries et al., 2019), **RobBERT** (v2) (Delobelle et al., 2022), and **RobBERTje** (non-shuffled) (Delobelle et al., 2021). BERTje is built upon the BERT architecture trained on 12GB of Dutch texts containing 2.4B tokens. RobBERT on the other hand, is based on the RoBERTa architecture, which boosts BERT's efficacy by pre-training in batches on 36GB of Dutch texts containing 6.6B tokens. RoBERTje employs a DistilBERT architecture, derived from RobBERTje, while preserving comparable efficacy with fewer parameters by using knowledge distillation.

We conducted a machine learning experiment with two model configurations: (a) fine-tuning the model solely on the provided essay text as input, and (b) fine-tuning the model on both the essay text and the given prompt as input. The models were imported with the Hugging Face library with a Pytorch backend and implemented to perform a regression task.

More details on the specific computing infrastructure can be found in Appendix A. For the final regression layer, we employed a sigmoid activation function as a way of bounding the scalar values to enhance the training stability. These bounded values functioned as predicted quality scores. Consistent with common practice in essay scoring (Alikaniotis et al., 2016; Yang et al., 2020; Li et al., 2022), all quality scores were min-max normalized before training. These normalized scores, along with the predicted scores, were used to compute the mean squared error, which functioned as the training loss. After training, the predicted scores were reverted to the original scale.

In the second configuration, the assignment prompt was taken into account in addition to the essay texts for the prediction of quality scores. We hypothesized that prompt information would be important for the prediction of quality scores, as the essays in the training set and the test set were written for different assignments. To incorporate this information into the model, we encoded the prompt using the same transformer model as for the essay text (i.e., a shared encoder). Two additional cross-attention layers were added to model the relationship between essays and prompts in both directions. This is similar to the configuration proposed by Liu et al. (2019).

The hyperparameters are given in Appendix B. These were selected based on preliminary results on a held-out set, comprising 15% of essays randomly selected from the training set, which were omitted during training but used for model evaluation.

3.3 Experimental Setup

To evaluate the reliability of the quality scores predicted by the fine-tuned models, we ran a machine learning experiment with the following training and test splits: $\{1, 2\} \rightarrow 3$, $\{1, 3\} \rightarrow 2$, and $\{2, 3\} \rightarrow 1$. In each fold, the three pre-trained models were fine-tuned on essays coming from two assignments (e.g., 1 and 2) and were evaluated on essays coming from the remaining assignment (e.g., 3). We employed this setup to emulate a real-world assessment scenario where we would have an assignment for which we do not have any scores yet (e.g., 3) and for which we need to predict initial quality scores based on scores estimated for other assignments (e.g., 1 and 2).

It is crucial to note that, despite the scores being logit values derived from distinct assignments, there was no complication in joining them within the training set. This was possible because the assignments were very similar, each assessing the quality of argumentative writing.

3.4 Evaluation Metric

Because our objective was to establish the reliability of predicted quality scores, we utilized the **squared Pearson correlation** (Bi, 2003)

$$\rho^2_{\theta^{init},\theta^*} = \frac{\mathrm{Var}_{\theta^*}}{\mathrm{Var}_{\theta^{init}}} \tag{3}$$

to assess the reliability between the predicted initial quality scores θ_i^{init} and final quality scores θ_i^* for $i = \{1, \ldots, n\}$ the essays in the test set. The reliability can be interpreted as the proportion of variance of the predicted initial quality scores that is attributed to the final quality scores. The closer this ratio is to one, the higher the reliability.

3.5 Efficiency Simulation Study

After having fine-tuning and evaluated pre-trained models, we simulated the impact of integrating model predictions as initial quality scores in CJ assessments. For each train-test split, we selected the model and its configuration (i.e., essay text with or without prompt) that exhibited the highest reliability. Subsequently, we conducted simulations to compare CJ assessments under **two conditions**: a warm-start BTL model (our experimental condition, where initial quality scores were predicted using the best model) and a cold-start BTL model (our control condition, where initial quality scores were absent).

While likelihood-based techniques (Hunter, 2004) are typically employed for parameter estimation in the BTL model (Equation 1), we adopted a **Bayesian approach** to simulate CJ assessments with both cold-start and warm-start BTL models. Within this framework, we could establish prior assumptions about the distribution of quality scores. Bayes' theorem allowed us to integrate these priors with judgments in the BTL model, resulting in posterior distributions for all quality scores. Compared to maximum likelihood estimation, Bayesian inference provides more stable estimates and a clearer understanding of the associated uncertainty (Phelan and Whelan, 2017).

3.5.1 Cold-Start Bayesian BTL Model

Under the cold-start condition, we formulated for each quality score a normal prior distribution (Equation 4) having a mean of 0 for all quality scores.

$$\theta_i \sim \mathrm{Normal}\left(0, \sigma_i^2\right) \tag{4}$$

This prior serves to regularize the distribution of quality scores, rendering it weakly informative. The lack of specificity about the essays for which quality scores are estimated characterizes this Bayesian BTL model as having a 'cold start'.

For the variance of each quality score, we specified a normal-truncated prior distribution (Equation 5), which is a common choice for $\sigma_i^2 \in (0, \infty)$.

$$\sigma_i^2 \sim \mathrm{Normal}_{Trunc}\left(\mu_0, \sigma_0^2\right) \tag{5}$$

The parameters of the distribution of σ_i^2 determined the level of uncertainty of the prior quality scores: the larger the location and scale parameters, the greater the prior uncertainty of the quality scores. Based on preliminary results, we chose to fix these parameters for all quality scores: $\mu_0 = 0.5$ and $\sigma_0^2 = 0.1$.

3.5.2 Warm-Start Bayesian BTL Model

Under the warm-start condition, we formulated prior distributions for the quality scores using the predicted quality scores. These priors are deemed informative, as they incorporate information about each essay's quality score.

To construct informative priors, we assumed a normal prior distribution for all quality scores θ_i for $i = \{1, \ldots, n\}$ with as mean their predicted initial quality scores θ_i^{init}.

$$\theta_i \sim \text{Normal}\left(\theta_i^{init}, \sigma_i^2\right) \tag{6}$$

All predicted quality scores were first centered, $\theta_i^{init} - \sum_{i=1}^{n} \theta_i^{init}$, to speed up convergence and encourage $\sum_{i=1}^{n} \theta_i \approx 0$. As in the cold-start condition, prior distributions were specified for the variance of the quality scores, measuring the uncertainty of the estimates (see Equation 5).

3.5.3 Sampling and Simulations

To estimate the posterior distribution of each θ_i and σ_i^2, samples were drawn according to the Hamiltonian Monte Carlo algorithm using Stan (Gelman et al., 2015), with 4 chains of 2000 steps of which 500 were warm-up steps. These were sufficient to reach convergence as diagnosed by a r-hat value of 1 (Vehtari et al., 2021). After convergence, the averages of the posterior distributions were used as point estimates.

To simulate a CJ assessment, we repeatedly estimated θ_i and σ_i^2 using increasingly more judgments; for an example of a simulated CJ assessment, see Appendix C. To account for possible effects of the order of judgments, we shuffled the sequence of judgments twenty times, resulting in twenty simulations of a CJ assessment. We repeated this process for each assessment, employing both a cold and a warm start.

3.5.4 Measuring Efficiency Gain

We assessed the gain in efficiency when introducing a warm start by observing the decrease in the average number of judgments required per essay to achieve a specific reliability level. The reliability of the quality scores was determined by the squared Pearson correlation (ρ^2_{θ,θ^*}) between the final quality scores θ^*, estimated at the end of the assessment, and the quality scores in a Bayesian BTL model estimated at a certain point during the assessment θ.

However, the use of this reliability metric presents a practical challenge. In practice, the reliability cannot be calculated during an assessment, as the final quality scores that would be estimated at the end of the assessment are still unknown. Hence, the reliability has to be approximated based on the estimated quality scores, which can be achieved using the Scale Separation Reliability (SSR$_\theta$). More specifically, the SSR$_\theta$ estimates Var$_{\theta^*}$ in Equation 3 by Var$_\theta - \mathbb{E}_{\sigma^2}$; see Equation 7. For a detailed derivation of the SSR$_\theta$, please refer to Verhavert et al. (2018). Note that we adjusted the reliability of the estimated quality scores to account for the reliability level of the final quality scores; see Appendix D.

$$\text{SSR}_\theta = \frac{\text{Var}_\theta - \mathbb{E}_{\sigma^2}}{\text{Var}_\theta} \to \rho^2_{\theta,\theta^*} \tag{7}$$

4 Results

4.1 Machine Learning Experiment

Table 3 shows the results of the machine learning experiment. The findings indicate that all fine-tuned language models effectively predicted quality scores for a completely new assignment, with correlation coefficients significantly different from zero. Notably, RobBERT consistently exhibited the highest reliability in predicting quality scores, aligning with its superior performance over other Dutch transformer models in diverse tasks (Delobelle et al., 2022).

Furthermore, when integrating both essay and prompt information, the RobBERT model consistently achieved the highest reliability with true quality scores. This observation aligns with previous AES research, emphasizing the predictive accuracy of essay scores across various prompts (Li et al., 2020; Do et al., 2023). As a result of these findings, we opted for the RobBERT model incorporating additional prompt information to predict initial quality scores in the simulation study.

It is crucial to note, however, that despite achieving high reliability, the fact that the reliability levels did not surpass 0.70 underscores the importance of assessor judgments to further improve the reliability of essay quality scores.

4.2 Simulation of CJ Assessments

The simulation study results, shown in Figure 3, highlight the comparison between CJ assessments under warm-start and cold-start conditions. The outcomes indicate that adopting a warm-start approach proved more efficient in terms of the number of judgments per essay needed to achieve a reliability level of at least 0.70.

In both Assignment 1 (Figure 3.c) and Assignment 3 (Figure 3.a), the desired reliability was

Fold	Essay Texts			+ Prompt Information		
	BERTje	**RobBERT**	**RobBERTje**	**BERTje**	**RobBERT**	**RobBERTje**
$\{1,2\} \rightarrow 3$	0.56	0.61	0.54	0.60	**0.63**	0.52
$\{1,3\} \rightarrow 2$	0.51	0.55	0.43	0.50	**0.59**	0.45
$\{2,3\} \rightarrow 1$	0.43	0.56	0.16	0.42	**0.57**	0.17
Average	0.50	0.57	0.38	0.52	**0.59**	0.37

Table 3: Squared Pearson correlations computed on the test set, comparing final quality scores and scores predicted by fine-tuned models, utilizing either only the essay texts or the prompts as well. Maximum scores are boldfaced.

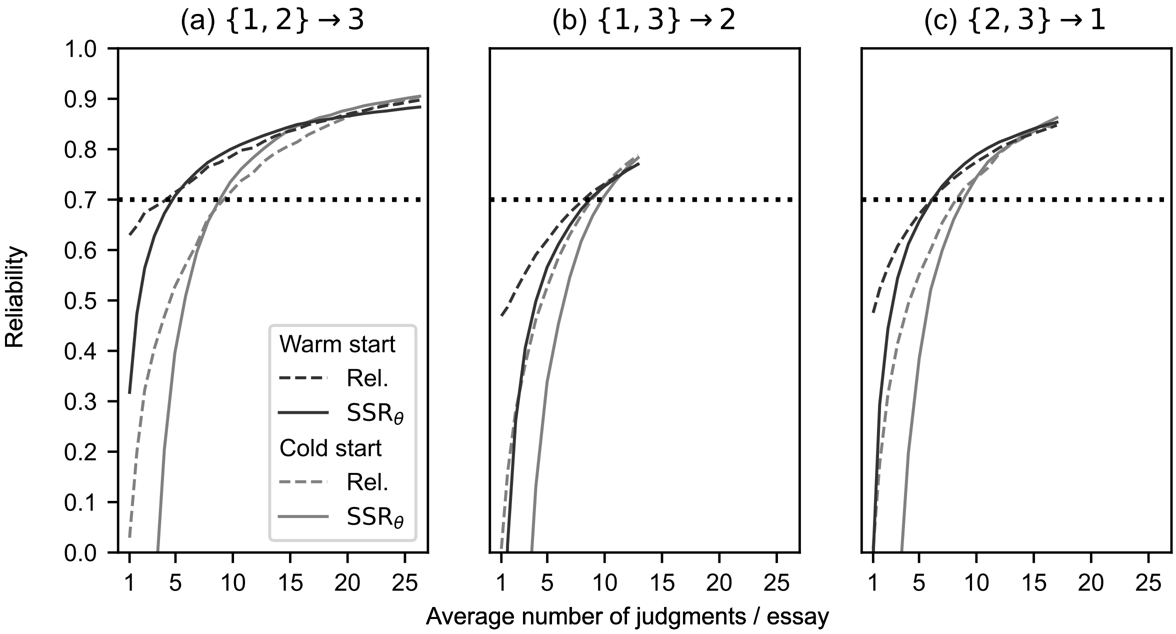


Figure 3: Results of simulated CJ assessments with a warm and a cold start. The average reliability and the average SSR$_\theta$ of the estimated quality scores are given in function of the average number of comparisons made per essay. These scores are averaged over 20 different orders of comparative judgments used to simulate an assessment.

reached with fewer than six judgments per essay. Conversely, employing a cold-start method required more than nine judgments per essay to attain an equivalent reliability level. Consequently, the warm-start approach resulted in efficiency gains of 35% and 41%, respectively. For Assignment 2 (Figure 3.b), a reliability of 0.70 required less than nine judgments per essay, while with a cold start at least ten judgments per essay were needed, which corresponds to an efficiency gain of 15%.

When exceeding ten judgments per essay, the disparity in reliability between warm and cold starts decreased across all assignments. This can be attributed to the diminishing impact of prior distributions on posterior distributions as the number of judgments increases. Additionally, for assignments 2 and 3, the reliability with a warm start begins to slightly trail behind that of the cold-start condition after ten judgments per essay. We posit that this observed difference may be associated with the choice to estimate final quality scores using a Bayesian BTL model with a cold start.

In practical scenarios, reliability is not accessible during assessments, making accurate measurement with the SSR$_\theta$ crucial. As shown in Figure 3, the SSR$_\theta$ demonstrated a faster approximation of reliability when employing a warm start compared to a cold start. Specifically, the SSR$_\theta$ reached the 0.70 reliability level for all assignments under a warm start. In contrast, the SSR$_\theta$ approached reliability at levels of 0.75 for Assignment 2 and 0.80 for Assignments 1 and 3 under a cold start.

To examine the impact of warm-starting assessments on *individual* quality scores, we compared the progression of quality score rankings. For illustration purposes, we show the results of one simulated assessment for Assignment 3. Figure 4 demonstrates that adopting a warm start led to qual-

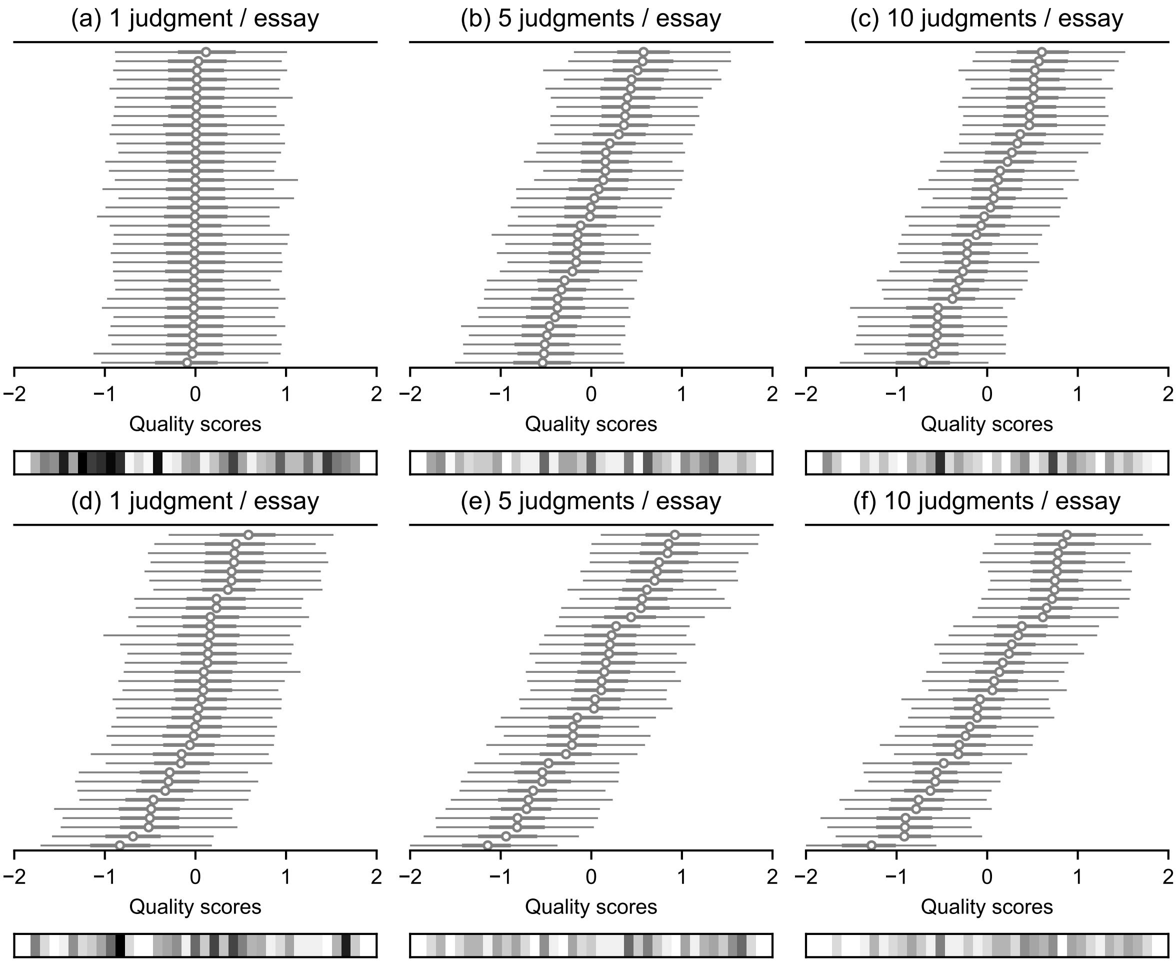


Figure 4: Forest plots of quality scores with 94%-high density intervals estimated at different stages of the CJ assessment of Assignment 3, with a cold-start condition (plots a–c above) and a warm-start condition (plots d–f below). The bar plots at the bottom show ranking accuracy based on the absolute differences in rank order of estimated and final quality scores, with darker shades indicating more incorrect rankings of estimated quality scores.

ity scores being more spread out, yielding a fairly accurate ranking at the start of the assessment. In contrast, quality scores under the cold-start condition clustered around the mean value, resulting in less precise rankings. This highlights the efficacy of informative priors in the warm-start condition in discerning between quality scores. Even after ten judgments per essay, the warm-start approach displayed a wider range of quality scores and a better ranking compared to the cold-start method.

5 Discussion

Our findings underscore the ability of current deep learning models, particularly transformer models, to predict initial quality scores that provide valuable information on the argumentative writing quality of essays. Furthermore, incorporating the assignment prompts for fine-tuning enhances the reliability of predicted quality scores, which aligns with prior research in AES (Li et al., 2020; Do et al., 2023; Sun et al., 2022b). We posit that prompt information is especially important for the prediction of initial quality score, since, in this study, the essays in the training set were written for different assignments than the essays in the test set.

When warm-starting CJ assessments with these predicted initial quality scores, the necessary number of comparative judgments to obtain reliable quality scores decreases significantly. This suggests that less effort from assessors is required while upholding high levels of reliability of the quality scores. Furthermore, our approach to increase the efficiency of CJ avoids any undesirable effects with respect to the reliability measures, which have been noted when employing an adaptive selection rule (Bramley, 2015; Bramley and

Vitello, 2019; Crompvoets et al., 2020). Additionally, our method demonstrates a more substantial improvement in efficiency compared to the approach of De Vrindt et al. (2022), who devised a more efficient selection rule based on similarities in essay texts.

6 Conclusion

We successfully improved the efficiency of CJ assessments by introducing a warm start for the estimation of the quality scores. This involved predicting essay quality scores, which were then used to form informative prior distributions within a Bayesian BTL model. Through an extensive simulation study, we demonstrated that our approach led to a reduction, ranging between 15% and 41%, in the number of comparative judgments needed to reach a reliability of 0.70 and produced more accurate rankings of essays at the start of an assessment. Furthermore, our findings indicate that these efficiency gains can be measured in practical settings, as the SSR_θ approximates the reliability well.

7 Limitations

To fine-tune the transformer models for the prediction of quality scores, we devised a training set combining the quality scores from different CJ assessments. This was feasible, as the quality scores measured the same quality of argumentative writing. However, if the essays were written in different text genres, such as informative writing, combining the quality scores would become non-trivial, since they measure a different kind of writing quality. Therefore, we recommend that before combining quality scores, they are first calibrated on a fixed scale using, for example, the method of Fair Averages (Linacre, 1989). Furthermore, differences in the genre of essays in train and test could make predicting the initial scores more difficult, causing lower reliability.

In this study, we assumed that the quality scores of essays written for other assignments were available to train a deep learning model for score prediction. However, settings may arise where these quality scores are unavailable, particularly in educational contexts where privacy concerns may prevent the inclusion of students' essays in a training set. In such cases, alternative methods for predicting scores must be explored. One approach is to train a deep learning model on publicly available AES datasets, such as the Automated Student Assessment Prize (ASAP) dataset published by the Hewlett Foundation (Hamner et al., 2012). However, it should be noted that these essays are written in English, prompting the need to evaluate how well a model trained on these can predict scores for Dutch essays. Alternatively, in case no essay scores are available for training, unsupervised learning approaches for AES could be considered (Ridley et al., 2020; Zhang and Litman, 2021).

To simulate the CJ assessments, we chose to repeatedly shuffle the order of judgments (see Appendix C). However, this approach may not reflect a realistic CJ assessment process, as, typically, pairs of essays for judgment are selected in such a way that each essay is compared (close to) the same number of times. For example, if an essay is compared 9 times and the others 10, that essay is selected and paired with a randomly selected essay. Based on preliminary results, we observed that our choice to repeatedly shuffle judgments has a negligible impact on the reliability results, as outlined in this study.

The current study reports an increase in reliability at the start of the assessment, but after more judgments have been made, the difference in reliability between a cold and a warm start became minimal (see Figure 3). For future research, we recommend exploring methods that use essay texts for the selection of pairs in a way that increases the reliability toward the end of an assessment, while avoiding the perverse effects that adaptive selection rules introduce (Bramley, 2015; Bramley and Vitello, 2019; van Daal et al., 2017).

Acknowledgements

This research was supported by a Baekeland mandate (HBC.2022.0164) granted by the Flanders Innovation & Entrepreneurship (VLAIO) to the first author, in cooperation with the company Comproved.

References

Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. 2016. Automatic text scoring using neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 715–725, Berlin, Germany. Association for Computational Linguistics.

Jian Bi. 2003. Agreement and reliability assessments for performance of sensory descriptive panel. *Journal of Sensory Studies*, 18(1):61–76.

Sue Bloxham. 2009. Marking and moderation in the uk: false assumptions and wasted resources. *Assessment & Evaluation in Higher Education*, 34(2):209–220.

Renske Bouwer, Elke Van Steendam, and Marije Lesterhuis. 2023. Guidelines for the validation of writing assessment in intervention studies. In Fien De Smedt, Renske Bouwer, Teresa Limpo, and Steve Graham, editors, *Conceptualizing, Designing, Implementing, and Evaluating Writing Interventions*, volume 40, pages 199–223. Brill.

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 Bramley. 2015. Investigating the reliability of adaptive comparative judgment. Technical report, University of Cambridge.

Tom Bramley and Sylvia Vitello. 2019. The effect of adaptivity on the reliability coefficient in adaptive comparative judgement. *Assessment in Education: Principles, Policy & Practice*, 26(1):43–58.

Elise Anne Victoire Crompvoets, Anton A Béguin, and Klaas Sijtsma. 2020. Adaptive pairwise comparison for educational measurement. *Journal of Educational and Behavioral Statistics*, 45(3):316–338.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. BERTje: A Dutch BERT Model. *arXiv preprint arXiv:1912.09582*.

Michiel De Vrindt, Wim Van den Noortgate, and Dries Debeer. 2022. Text mining to alleviate the cold-start problem of adaptive comparative judgments. *Frontiers in Education*, 7:132–147.

Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2020. RobBERT: a Dutch RoBERTa-based Language Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3255–3265, Online. Association for Computational Linguistics.

Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2021. Robbertje: A distilled dutch bert model. *Computational Linguistics in the Netherlands Journal*, 11:125–140.

Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2022. Robbert-2022: Updating a dutch language model to account for evolving language use. *arXiv preprint arXiv:2211.08192*.

Heejin Do, Yunsu Kim, and Gary Geunbae Lee. 2023. Prompt- and trait relation-aware cross-prompt essay trait scoring. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1538–1551, Toronto, Canada. Association for Computational Linguistics.

Explosion. 2023. Available trained pipelines for dutch: nl_core_news_sm. `https://spacy.io/models/nl#nl_core_news_sm` [Accessed on April 3, 2024)].

Andrew Gelman, Daniel Lee, and Jiqiang Guo. 2015. Stan: A probabilistic programming language for bayesian inference and optimization. *Journal of Educational and Behavioral Statistics*, 40(5):530–543.

Ben Hamner, Jaison Morgan, Mark Shermis lynnvandev, and Tom Vander Ark. 2012. The hewlett foundation: Automated essay scoring.

David R Hunter. 2004. MM algorithms for generalized Bradley-Terry models. *The Annals of Statistics*, 32(1):384–406.

Beata Beigman Klebanov and Nitin Madnani. 2022. *Automated Essay Scoring*. Number 52 in Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, San Rafael.

Donald Laming. 2003. *Human judgment: The eye of the beholder*. Cengage Learning, London, United Kingdom.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*.

Marije Lesterhuis, Renske Bouwer, Tine van Daal, Vincent Donche, and Sven De Maeyer. 2022. Validity of comparative judgment scores: How assessors evaluate aspects of text quality when comparing argumentative texts. *Frontiers in Education*, 7:122–131.

Xia Li, Minping Chen, and Jian-Yun Nie. 2020. Sednn: Shared and enhanced deep neural network model for cross-prompt automated essay scoring. *Knowledge-Based Systems*, 210:106491.

Xia Li, Huali Yang, Shengze Hu, Jing Geng, Keke Lin, and Yuhai Li. 2022. Enhanced hybrid neural network for automated essay scoring. *Expert Systems*, 39(10):e13068.

John Michael Linacre. 1989. *Many-faceted Rasch measurement*. Ph.D. thesis, The University of Chicago.

Tiaoqiao Liu, Wenbiao Ding, Zhiwei Wang, Jiliang Tang, Gale Yan Huang, and Zitao Liu. 2019. Automatic short answer grading via multiway attention networks. *arXiv preprint arXiv:1909.10166*.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*.

R. Duncan Luce. 1959. On the possible psychophysical laws. *Psychological Review*, 66(2):81–95.

Arya D. McCarthy, Kevin P. Yancey, Geoffrey T. LaFlair, Jesse Egbert, Manqian Liao, and Burr Settles. 2021. Jump-starting item parameters for adaptive language tests. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 883–899, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Suzanne McMahon and Ian Jones. 2015. A comparative judgement approach to teacher assessment. *Assessment in Education: Principles, Policy & Practice*, 22(3):368–389.

Farjana Sultana Mim, Naoya Inoue, Paul Reisert, Hiroki Ouchi, and Kentaro Inui. 2019. Unsupervised learning of discourse-aware text representation for essay scoring. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 378–385, Florence, Italy. Association for Computational Linguistics.

Christopher M Ormerod, Akanksha Malhotra, and Amir Jafari. 2021. Automated essay scoring using efficient transformer-based language models. *arXiv preprint arXiv:2102.13136*.

Gustavo Penha and Claudia Hauff. 2020. What does bert know about books, movies and music? probing bert for conversational recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems*, RecSys '20, page 388–397, New York, NY, USA. Association for Computing Machinery.

Gabriel C Phelan and John T Whelan. 2017. Hierarchical bayesian bradley-terry for applications in major league baseball. *arXiv preprint arXiv:1712.05879*.

Konstantinos Pliakos, Seang-Hwane Joo, Jung Yeon Park, Frederik Cornillie, Celine Vens, and Wim Van den Noortgate. 2019. Integrating machine learning into item response theory for addressing the cold start problem in adaptive learning systems. *Computers & Education*, 137:91–103.

Alastair Pollitt. 2012. The method of adaptive comparative judgement. *Assessment in Education: Principles, Policy & Practice*, 19(3):281–300.

Dadi Ramesh and Suresh Kumar Sanampudi. 2022. An automated essay scoring systems: a systematic literature review. *Artificial Intelligence Review*, 55(3):2495–2527.

Robert Ridley, Liang He, Xinyu Dai, Shujian Huang, and Jiajun Chen. 2020. Prompt agnostic essay scorer: A domain generalization approach to cross-prompt automated essay scoring. *arXiv preprint arXiv:2008.01441*.

Burr Settles, Geoffrey T. LaFlair, and Masato Hagiwara. 2020. Machine learning–driven language assessment. *Transactions of the Association for Computational Linguistics*, 8:247–263.

Geng Sun, Wei Wei, Tingru Cui, Dongming Xu, Shiping Chen, Alex Shvonski, Li Li, Jun Shen, and Soheila Garshasbi. 2022a. Adapting new learners and new resources to micro open learning via online computation. *IEEE Transactions on Computational Social Systems*, 9(6):1807–1819.

Jingbo Sun, Tianbao Song, Jihua Song, and Weiming Peng. 2022b. Improving automated essay scoring by prompt prediction and matching. *Entropy*, 24(9):1–15.

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.

Masaki Uto, Yikuan Xie, and Maomi Ueno. 2020. Neural automated essay scoring incorporating handcrafted features. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6077–6088, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Tine van Daal, Marije Lesterhuis, Liesje Coertjens, Vincent Donche, and Sven De Maeyer. 2016. Validity of comparative judgement to assess academic writing: examining implications of its holistic character and building on a shared consensus. *Assessment in Education Principles Policy and Practice*, 26:59–74.

Tine van Daal, Marije Lesterhuis, Liesje Coertjens, Marie-Thérèse van de Kamp, Vincent Donche, and Sven De Maeyer. 2017. The complexity of assessing student work using comparative judgment: The moderating role of decision accuracy. *Frontiers in Education*, 2:1–13.

Aki Vehtari, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Bürkner. 2021. Rank-normalization, folding, and localization: An improved r^ for assessing convergence of mcmc (with discussion). *Bayesian Analysis*, 16(2).

San Verhavert, Renske Bouwer, Vincent Donche, and Sven De Maeyer. 2019. A meta-analysis on the reliability of comparative judgement. *Assessment in Education: Principles, Policy & Practice*, 26(5):541–562.

San Verhavert, Sven De Maeyer, Vincent Donche, and Liesje Coertjens. 2018. Scale separation reliability: what does it mean in the context of comparative judgment? *Applied Psychological Measurement*, 42(6):428–445.

Cong Wang, Zhiwei Jiang, Yafeng Yin, Zifeng Cheng, Shiping Ge, and Qing Gu. 2023. Aggregating multiple heuristic signals as supervision for unsupervised automated essay scoring. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13999–14013, Toronto, Canada. Association for Computational Linguistics.

Jianling Wang, Haokai Lu, James Caverlee, Ed Chi, and Minmin Chen. 2024. Large language models as data augmenters for cold-start item recommendation. *arXiv preprint arXiv:2402.11724*.

Ruosong Yang, Jiannong Cao, Zhiyuan Wen, Youzheng Wu, and Xiaodong He. 2020. Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1560–1569, Online. Association for Computational Linguistics.

Haoran Zhang and Diane Litman. 2021. Essay Quality Signals as Weak Supervision for Source-based Essay Scoring. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 85–96, Online. Association for Computational Linguistics.

A Computing Information

We implemented both transformer models for quality score prediction using Pytorch 2.1.0, Hugging Face 4.32.1, and Python 3.9.12. We conducted the experiments on a system running Ubuntu 22.04.2.

B Hyperparameters

The AdamW optimizer was used (Loshchilov and Hutter, 2017), with a polynomial learning rate scheduler and a starting learning rate of $1e-5$. The warm-up ratio was set at 10% of the steps, with a batch size of 5. The weight decay was set to 0.09. Furthermore, a 5% dropout was used to prevent overfitting. The transformer models were fine-tuned for 40 epochs with the possibility of early stopping based on the evaluation metric measured on the held-out set.

C Example of Simulated CJ Assessment

For the CJ assessment of Assignment 3, 27 judgments were made for each essay, as detailed in Table 2. This means that each essay was involved in 27 pairwise comparisons. Given that there are 35 essays part of the assessment, assessors had to make $35 \times 27/2 \approx 473$ judgments in total. To simulate the CJ assessment of Assignment 3, all θ_i and σ_i^2 parameters in a Bayesian BTL model were iteratively estimated using 1 to 473 judgments. Following each estimation, the SSR_θ and reliability were computed. Recognizing that the order of judgments selected could influence the estimates and reliability levels, we shuffled the sequence of judgments twenty times and repeated the procedure mentioned above.

D Adjusting the Reliability Measure

In studies on the reliability of CJ, the 'true quality scores' are obtaining using a all-play-all design (Bramley, 2015; Crompvoets et al., 2020), where every pairwise combination essays has been judged. Since the data in this study were not gathered using an all-play-all, we assume that the final quality scores are, in fact, the true scores. However, these final quality scores possess their own level of reliability, as given by the SSR of the estimated quality scores at the end of a CJ assessment: $\text{SSR}_{\theta*}$. To account for this, we adjusted the reliability of the estimated quality scores, $\rho^2_{\theta,\theta*}$, by multiplying it by $\text{SSR}_{\theta*}$. Consequently, SSR_θ converges to $\text{SSR}_{\theta*}$, when the estimated quality scores align with the final quality scores at the end of the assessment (i.e., when $\rho^2_{\theta,\theta*} \approx 1$).

Improving Transfer Learning for Early Forecasting of Academic Performance by Contextualizing Language Models

Ahatsham Hayat[1], **Bilal Khan**[2], **Mohammad Rashedul Hasan**[1]
University of Nebraska-Lincoln[1], Lehigh University[2]
aahatsham2@huskers.unl.edu, bik221@lehigh.edu, hasan@unl.edu

Abstract

This paper presents a cutting-edge method that harnesses contextualized language models (LMs) to significantly enhance the prediction of early academic performance in STEM fields. Our approach uniquely tackles the challenge of transfer learning with limited-domain data. Specifically, we overcome this challenge by contextualizing students' cognitive trajectory data through the integration of both distal background factors (comprising academic information, demographic details, and socioeconomic indicators) and proximal non-cognitive factors (such as emotional engagement). By tapping into the rich prior knowledge encoded within pre-trained LMs, we effectively reframe academic performance forecasting as a task ideally suited for natural language processing.

Our research rigorously examines three key aspects: the impact of data contextualization on prediction improvement, the effectiveness of our approach compared to traditional numeric-based models, and the influence of LM capacity on prediction accuracy. The results underscore the significant advantages of utilizing larger LMs with contextualized inputs, representing a notable advancement in the precision of early performance forecasts. These findings emphasize the importance of employing contextualized LMs to enhance artificial intelligence-driven educational support systems and overcome data scarcity challenges.

1 Introduction

Modern artificial intelligence (AI) methods, such as deep learning (DL), have increasingly been deployed as cost-effective solutions to develop early-warning systems across various sectors, including health (Adler et al., 2022; Mamun et al., 2022; Zhao et al., 2019; Horwitz et al., 2022; Liu et al., 2023a; Collins et al., 2023; Xu et al., 2023; Adler et al., 2020) and education (Wang et al., 2016, 2014; Li et al., 2020; Xu and Ouyang, 2022). These systems leverage forecasting-based interventions to preemptively address potential issues, from medical conditions to academic performance. In the educational domain, specifically, AI-based interventions utilize cognitive data, like students' course-related assessment scores, to predict and improve academic outcomes (Greenstein et al., 2021; Arnold and Pistilli, 2012; Liu et al., 2023b). The efficacy of these interventions hinges on the precision of early forecasts—predicting course performance as early as possible (Hasan and Aly, 2019; Hasan and Khan, 2023). However, this poses a significant challenge when training data is scarce, leading to suboptimal model performance. Transfer learning could offer a solution, yet the approach is hampered by the lack of relevant pre-trained models or sufficiently large, domain-specific datasets for pre-training (Tsiakmaki et al., 2020).

In this paper, we address the challenges associated with limited training data by introducing a novel transfer learning methodology specifically tailored for domain-specific data within STEM (Science, Technology, Engineering, and Mathematics) education contexts. We propose leveraging Transformer-based (Vaswani et al., 2017) pre-trained language models (LMs) for early prediction of academic performance in undergraduate STEM courses. Our method exploits the extensive knowledge base (Raffel et al., 2020; Roberts et al., 2020) and reasoning capabilities (Chowdhery et al., 2022; Wei et al., 2023; Bhatia et al., 2023) of LMs, transforming end-of-the-semester performance forecasting into a natural language text generation task.

To enhance knowledge transfer using limited domain data, we **contextualize** students' cognitive data by integrating both distal background factors and proximal non-cognitive factors. This multidimensional approach encompasses demographic, socioeconomic, and academic background factors, as well as non-cognitive features like emotional engagement, to enrich the predictive model. By transforming the ordinal (numeric or real-valued) features of our data into natural language text sequences, we tailor pre-trained LMs to our specific task. Additionally, we augment these sequences to increase the dataset size, thereby improving predictive accuracy through a more balanced representation of various performance outcomes.

Contextualizing Academic Trajectories. Our approach integrates students' background and engagement data to provide a comprehensive view of their academic journey. Based on Social Cognitive Career Theory (Bandura, 2001), we hypothesize that a student's course performance correlates with their background, suggesting that LMs can learn individualized academic patterns. Furthermore, longitudinal non-cognitive data, reflecting aspects like motivation and engagement, are posited to have a strong correlation with students' academic trajectories, potentially enhancing the LMs' predictive accuracy (Fogg, 2009; Fredricks, 2014).

Our contextualization process divides into four categories:

- **Demographic Contextualization**: Includes inherent personal and social identity factors, such as race and gender. These are critical for understanding the diverse identities students bring to their educational experiences and how these aspects influence their academic outcomes in the course.

- **Socioeconomic Contextualization**: Encompasses factors related to the economic status and background of the student's family, like parent's total yearly income. This contextualization helps to understand the resources and socio-economic pressures that might influence a student's academic performance and opportunities.

- **Academic Contextualization**: Pertains to the specifics of a student's educational path, including their class standing year (freshman, sophomore, junior, senior) and their chosen major. This type of contextualization is vital for understanding how students' educational choices and progression affect performance.

- **Emotional Engagement Contextualization**: Centers on students' emotional and perceptual dimensions of academic engagement. Specifically, it aims to explore how students' anticipations of academic outcomes (expected grades) and their satisfaction with their academic performance influence their engagement, motivation, and overall educational journey.

Using the contextualized academic trajectory data, we address the following research questions.

- **[RQ1]**: How does contextualization of academic trajectory data impact the efficacy of transfer learning from pre-trained LMs in early academic performance forecasting?

- **[RQ2]**: How does a natural language text generation approach compare with numeric feature-based models in early performance forecasting?

- **[RQ3]**: What impact does the capacity of pre-trained LMs (i.e., the number of parameters) have on forecasting accuracy?

Our primary contributions are threefold.

Innovative Methodology: We introduce a novel methodology that employs natural language text generation for the early forecasting of academic performance, showcasing a unique blend of linguistic and educational insights.

Contextualization as a Catalyst for Transfer Learning: We demonstrate that contextualizing academic trajectory data significantly enhances the transfer learning process from pre-trained LMs. By embedding both cognitive and non-cognitive features within a rich contextual narrative, our approach unlocks the vast potential of LMs to understand and predict academic outcomes with remarkable accuracy.

Exploitation of Pre-trained LM Knowledge: Our research underscores the pivotal role of leveraging the inherent, comprehensive knowledge encapsulated within LMs. Through our method, we illustrate how the nuanced understanding and versatility of LMs can be effectively harnessed for the domain-specific task of predicting student performance, thus marking a significant advancement in the field of educational AI.

The remainder of the paper is organized as follows: Section 2 outlines our methodology, encompassing a description of the dataset and its collection. In Section 3, we present the experiments and provide a detailed analysis of the results, followed by our conclusions and suggestions for future work in Section 4. Finally, Section 5 offers a discussion of pertinent literature.

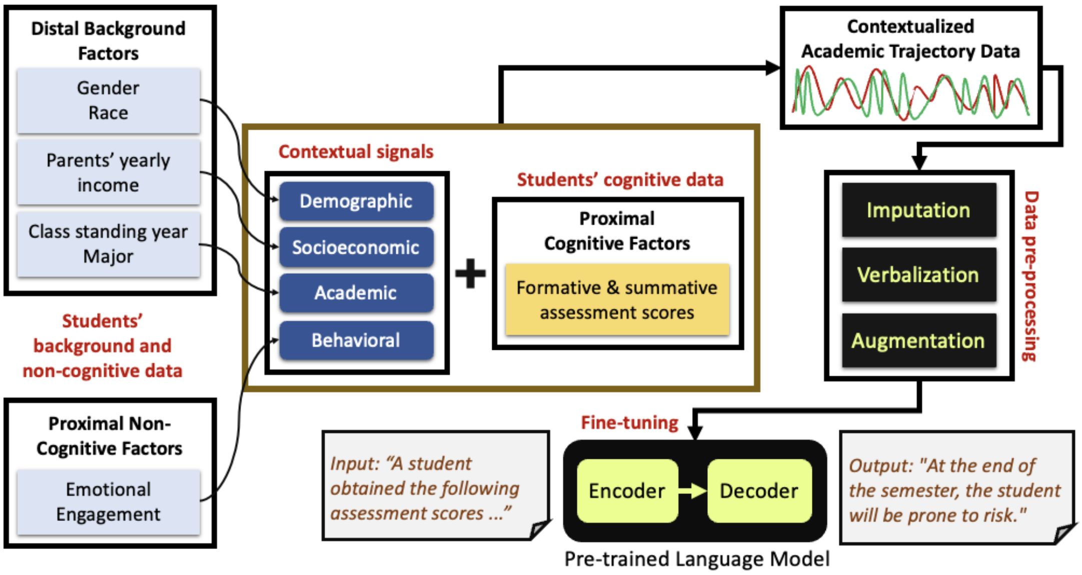

Distal Background Factors
Gender
Race
Parents' yearly income
Class standing year
Major
Students' background and non-cognitive data
Proximal Non-Cognitive Factors
Emotional Engagement
Contextual signals
Demographic
Socioeconomic
Academic
Behavioral
+
Students' cognitive data
Proximal Cognitive Factors
Formative & summative assessment scores
Contextualized Academic Trajectory Data
Imputation
Verbalization
Augmentation
Data pre-processing
Fine-tuning
Input: "A student obtained the following assessment scores ..."
Encoder
Decoder
Pre-trained Language Model
Output: "At the end of the semester, the student will be prone to risk."

Figure 1: An overview of the approach for enhancing transfer learning from pre-trained language models for early academic performance forecasting.

2 Method

To harness the nuanced understanding pre-trained LMs offer regarding students' academic experiences, we assembled a **detailed longitudinal dataset** that examines the interplay among various factors, including background, cognitive, and non-cognitive elements in student learning. Figure 1 illustrates the LM-based transfer learning framework, featuring the contextualization of proximal cognitive data followed by the preprocessing of the contextualized academic trajectory. Data contextualization involves integrating distal background and proximal non-cognitive factors with cognitive trajectory data. Below, we outline the process of compiling a language dataset, encompassing data collection and pre-processing methods, and conclude with a formal description of transfer learning through fine-tuning of LMs.

2.1 Data Collection

Our dataset comprises information obtained from 48 first-year college students enrolled in an introductory programming course at a public university in the United States, following approval from the University's Institutional Review Board. The dataset encompasses three key dimensions of the students' academic journeys.

Background Data (5-dimensional): At the outset of the semester, critical 5-dimensional background data was collected through a Qualtrics-based multiple-choice web survey. This numeric dataset includes students' academic details (such as class standing year and major), demographic information (including gender and race), and a socioeconomic indicator (family yearly income).

Non-Cognitive Data (2-dimensional): This dimension includes longitudinal measures of students' emotional engagement throughout the semester, comprising 2-dimensional data reflecting students' anticipated end-of-semester performance and their current performance satisfaction, both in numeric format.

The data is collected via a **privacy-preserving smartphone application**, designed to prompt contextually relevant, study-specific multiple-choice questions daily. This ensures that participants' anonymized responses are securely compiled on cloud servers for subsequent analysis. Each participant is assigned a unique randomly generated ID upon enrollment, with no personally identifiable information collected via the app. All data collected is tagged solely by the participant's random ID, with no linkage maintained between the ID and participant identity. Geolocation and Bluetooth sensors are utilized in the app to ascertain instantaneous context for question triggers, although sensor data is not persistently stored. By transparently informing students about the privacy-preservation mechanisms, we mitigate potential psychological and academic incentives for artificial performance or dishonest responses during experience sampling.

Furthermore, this privacy-preserving mechanism serves to mitigate potential biases in the data collection process. By anonymizing participants' responses and ensuring that no personally identifiable information is collected, we minimize the risk of participants feeling pressured to provide socially desirable responses. This approach promotes more authentic and unbiased data collection, contributing to the reliability and validity of our findings.

Cognitive Data (21-dimensional): The dataset also includes 21-dimensional numeric cognitive data derived from students' assessment scores (both formative and summative) over the first 8 weeks of the semester. This cognitive data was obtained from the course's learning management system, Canvas, providing insights into students' academic performance, engagement, and progress within the course curriculum.

2.2 Data Contextualization

We enriched students' cognitive trajectory data—comprising their course-related formative and summative scores—by incorporating four contextual dimensions: demographic (gender and race), academic (class standing year and major), socioeconomic (family yearly income), and behavioral (emotional engagement). The dynamic cognitive and non-cognitive data were intertwined to preserve their temporal sequence, while the static background data was added at the end of the trajectory.

2.3 Data Pre-processing

The contextualized numeric trajectory data underwent preprocessing to adapt it for LM use, which included handling missing values in the non-cognitive data, verbalization of the data, and data augmentation for enhanced model training.

Data Imputation. The proximal non-cognitive data exhibited missing values, resulting from participants either skipping questions or temporarily uninstalling the app. We encountered two distinct patterns of missing data: complete absence of responses for an entire day and partial absences within a day. To address days with entirely missing data, we employed the Last Observation Carried Forward (LOCF) imputation method (Liu, 2016). This method involves carrying forward the last observed value for each participant to replace missing values at subsequent time points. While LOCF is a commonly used approach due to its simplicity, it assumes that the missing data points would have followed a similar trajectory as the last observed value. In situations where no prior data were available, the Next Observation Carried Backward (NOCB) approach was employed (Jahangiri et al., 2023), using data from a subsequent day that contained relevant responses. The challenge of partially missing data, particularly for follow-up questions, necessitated a more nuanced approach. When the preceding day's trigger question response did not match, directly applying LOCF for the follow-up question was deemed unreliable (Lachin, 2016). Instead, we filled these gaps with responses from days where the trigger question responses aligned. If no matching previous day could be identified, a future day with corresponding answers was utilized.

Data Verbalization. To transform the numeric dataset into natural language, we designed a template for verbalizing both the input (X) and output (Y) data sequences (refer to the Appendix for details). Input sequences were prefaced with contextual messages, such as "A student obtained the following assessment scores in an introductory programming course ..." for cognitive data, and "Some background information about the student: ..." for distal data. Chronological order was emphasized by prefacing data with the week number, e.g., "*In week [WEEK_NUMBER]*". The output sequences, categorized into four performance groups (at-risk, prone-to-risk, average, outstanding), contextualized the final letter grade in a natural language expression, e.g., "*At the end of the semester, the student will be at risk.*" . This verbalization process yielded three datasets based on 8-week, 4-week, and 2-week long input sequences.

Data Augmentation. Given the initial dataset's unbalanced distribution across performance categories (24 instances of outstanding, 12 average, 6 prone-to-risk, and 6 at-risk), we employed a twofold approach for data augmentation. Firstly, we utilized oversampling techniques (Haixiang et al., 2017; Hernandez et al., 2013) to duplicate instances from minority classes, thus balancing the dataset. Secondly, we incorporated synonym replacement methods (Li et al., 2022), which involved substituting words with their synonyms to introduce token variations. This comprehensive approach aimed to not only address class imbalance but also enrich the dataset with diverse token variations.

As a result of our data augmentation strategy, the augmented dataset showcased a more equitable distribution among performance categories, totaling 144 samples, comprising 48 instances of outstanding, 36 average, 30 prone-to-risk, and 30 at-risk.

These methodologies provide a robust foundation for applying transfer learning to LMs, facilitating a deep understanding of students' academic performance through a multi-dimensional data lens.

2.4 Fine-tuning LMs

Each sequence in X and Y contains standard lexical literals used in English (e.g., words and phrases), which is used to fine-tune a pre-trained encoder-decoder LM. The encoder $f_E(.)$ maps the input sequence $(x_1, x_2, ..., x_l)$ to an intermediate latent embedding sequence $(z_1, z_2, ..., z_l)$.

$$z = f_E(x_1, x_2, ..., x_l; \theta_E) \tag{1}$$

where θ_E are the weights of the encoder.

The decoder $f_D(.)$ takes the latent embeddings $(z_1, z_2, ..., z_l)$ to generate an output sequence $(\hat{y}_1, \hat{y}_2, ..., \hat{y}_m)$ in an auto-regressive fashion, i.e., at each step the decoder $f_D(.)$ uses previously generated symbols $\hat{y}_{<m}$ as additional input for generating the next token $\hat{y}_m$.The probability of generating the m-th token $\hat{y}_m$ is given by

$$\begin{aligned} & p(\hat{y}_m|\hat{y}_{<m}; z_1, z_2, ..., z_l) \\ & \quad = softmax(f_D(\hat{y}_{<m}; z_1, z_2, ..., z_l; \theta_D)) \end{aligned} \tag{2}$$

where θ_D are the weights of the decoder. For fine-tuning the encoder-decoder LM, the multi-class cross-entropy loss function is used. The number of classes in the loss function is set by the total number of tokens in the vocabulary. For a batch size B, the loss function is:

$$\mathcal{L} = -\sum_{b=1}^{B}\sum_{m=1}^{M} y_m^b log \hat{y}_m^b \tag{3}$$

3 Experiments

To thoroughly investigate the research questions outlined in Section 1, we performed a series of experiments focusing on the learning capabilities of LMs. These experiments involved fine-tuning pre-trained LMs across multi-dimensional language datasets spanning 8 weeks, 4 weeks, and 2 weeks. This selection of timeframes facilitated an in-depth examination of LM adaptability over various periods. The effectiveness of the adapted LMs was assessed through their ability to identify performance types based on matching keywords in the predicted output sequences. Moreover, we explored the impact of LM size—small, medium, and large—on their performance.

Experimental Setup. For the encoder-decoder LM, we used pre-trained FLAN-T5 (Chung et al., 2022), which is a variant of the T5 model (Raffel et al., 2020). The FLAN-T5 model is instruction fine-tuned, making it suitable for our purposes. We employed FLAN-T5 with three different capacities, determined by the number of parameters: FLAN-T5-Small (80M), FLAN-T5-Base (250M), and FLAN-T5-Large (770M). These LMs have a context window limited to 512 tokens. As baseline comparisons, we utilized four models that work with only numeric features: three neural networks (NNs) and one non-NN machine learning model. The neural networks include a Long Short-Term Memory (LSTM) network (Hochreiter and Schmidhuber, 1997), a Convolutional Neural Network (CNN) with a one-dimensional (1D) convolutional kernel (Kim, 2014), and a Transformer network (Vaswani et al., 2017). The non-NN machine learning model employed was a Support Vector Machine (SVM) with a linear kernel (Boser et al., 1992), which demonstrated superior performance over the Gaussian Radial Basis Function kernel.

The baseline models were trained using 3 variably-length numeric datasets containing only the cognitive features. Exploring baseline models with all three feature types is planned as future work. To ensure compatibility with the LM-based experiments, the numeric datasets were created from the augmented verbalized datasets by decoding the cognitive feature part of text sequences into numeric values.

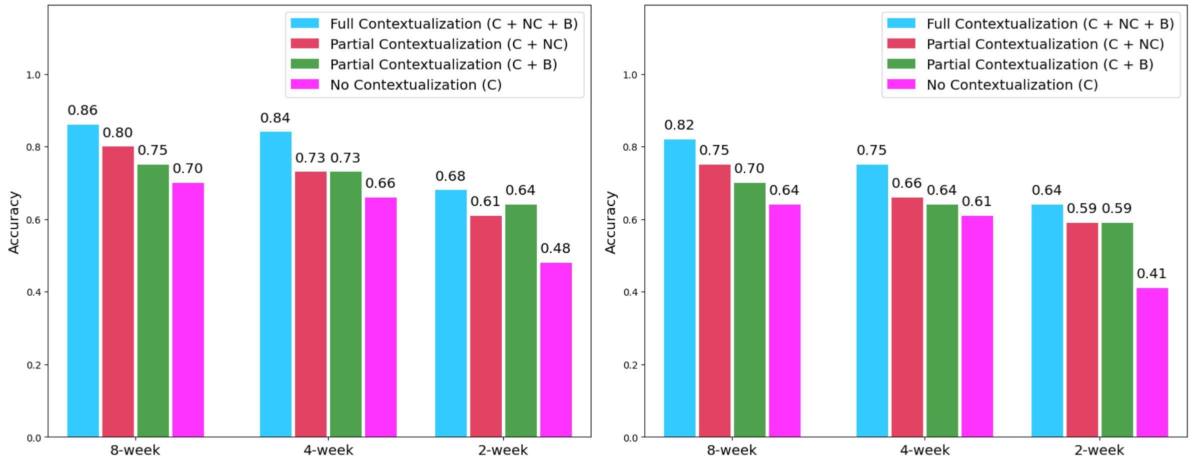
We used the same test sets to evaluate both model types, employing the following metrics: accuracy, precision, recall, and F1 score. A detailed description of the experimental setup is provided in the Appendix.

3.1 Results

[RQ1]: ***How does contextualization of academic trajectory data impact the efficacy of transfer learning from pre-trained LMs in early academic performance forecasting?*** The core objective of this study is to evaluate how the contextualization of academic trajectory data influences the forecasting effectiveness of pre-trained LMs. To this end, we fine-tuned LMs of varying sizes with aca-

Table 1: Evaluation of the large LM (FLAN-T5-Large) fine-tuned with four combinations of the 3 feature types using the 8-week, 4-week, and 2-week datasets. The best results are in **bold**.
Legends: C=Cognitive, NC=Non-Cognitive, B=Background, AR=At-Risk, PR=Prone-To-Risk, AV=Average, OU=Outstanding, P=Precision, R=Recall, F1=F1 Score, A=Accuracy

Features	Class	8-week				4-week				2-week			
		P	R	F1	A	P	R	F1	A	P	R	F1	A
Full Contextualization (C + NC + B)	AR	0.78	1.00	0.88	**0.89**	1.00	1.00	1.00	**0.84**	0.64	1.00	0.78	**0.77**
	PR	0.89	0.80	0.84		0.89	0.80	0.84		1.00	0.50	0.67	
	AV	0.92	1.00	0.96		0.71	0.91	0.80		0.73	1.00	0.85	
	OU	0.93	0.81	0.87		0.86	0.75	0.80		0.85	0.69	0.76	
Partial Contextualization (C + NC)	AR	0.70	1.00	0.82	0.82	0.70	1.00	0.82	0.77	0.62	0.71	0.67	0.68
	PR	1.00	0.60	0.75		0.86	0.60	0.71		0.71	0.50	0.59	
	AV	0.73	1.00	0.85		0.69	1.00	0.81		0.62	0.91	0.74	
	OU	0.92	0.75	0.83		0.91	0.62	0.74		0.77	0.62	0.69	
Partial Contextualization (C + B)	AR	0.78	1.00	0.88	0.77	0.88	1.00	0.93	0.77	0.60	0.86	0.71	0.64
	PR	0.89	0.80	0.84		0.71	1.00	0.83		0.71	0.50	0.59	
	AV	0.67	0.73	0.70		0.69	0.82	0.75		0.70	0.64	0.67	
	OU	0.79	0.69	0.73		0.89	0.50	0.64		0.59	0.62	0.61	
No Contextualization (C)	AR	0.60	0.86	0.71	0.73	0.62	0.71	0.67	0.70	0.36	0.57	0.44	0.52
	PR	0.86	0.60	0.71		0.67	0.60	0.63		0.88	0.70	0.78	
	AV	0.60	0.82	0.69		0.67	0.91	0.77		0.54	0.64	0.58	
	OU	0.92	0.69	0.79		0.83	0.62	0.71		0.42	0.31	0.36	



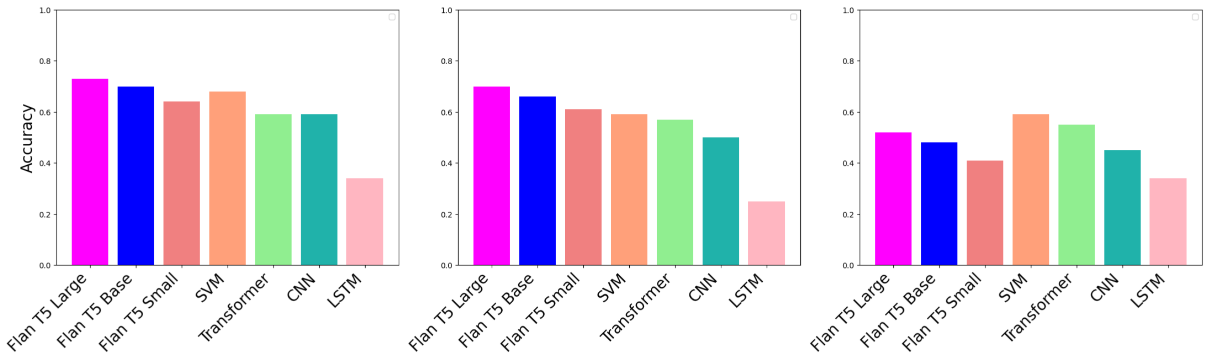
(a) FLAN-T5 Base

(b) FLAN-T5 Small

Figure 2: Impact of contextualization on the FLAN-T5 Base and Small models.

demic trajectory data enriched with three types of features: cognitive (C), non-cognitive (NC), and background (B). This investigation includes comparing the performance impact between fully contextualized LMs (utilizing all three feature types) and partially-contextualized or non-contextualized LMs. For partial contextualization, we explored combinations of C+NC and C+B features, whereas, in the non-contextualization scenario, only cognitive (C) features were employed for model fine-tuning.

According to the performance metrics provided in Table 1 for the best-performing large LM, FLAN-T5-Large, it is evident that models utilizing a contextualization approach, whether fully or partially, significantly outperform those without any contextualization. Specifically, the **fully contextualized LMs demonstrate superior forecasting abilities**. For instance, such a model can predict student performance with an accuracy of 77% by the end of the 2nd week of the semester. This early prediction capability is vital for implementing effective early



(a) 8-week

(b) 4-week

(c) 2-week

Figure 3: Comparison with baseline models on cognitive features.

Table 2: Evaluation of the three baseline models trained with cognitive features using the 8-week, 4-week, and 2-week datasets. The best results are in **bold**.
Legends: AR=At-Risk, PR=Prone-To-Risk, AV=Average, OU=Outstanding, P=Precision, R=Recall, F1=F1 Score, A=Accuracy

Model	Class	8-week				4-week				2-week			
		P	R	F1	A	P	R	F1	A	P	R	F1	A
CNN	AR	0.50	0.86	0.63	0.59	0.44	0.57	0.50	0.50	0.45	0.71	0.56	0.45
	PR	0.83	0.50	0.62		1.00	0.30	0.46		0.44	0.70	0.54	
	AV	1.00	0.09	0.17		0.33	0.55	0.43		0.22	0.18	0.20	
	OU	0.56	0.88	0.68		0.37	0.56	0.58		0.75	0.38	0.50	
LSTM	AR	1.00	0.14	0.25	0.34	0.00	0.00	0.00	0.25	0.15	0.29	0.20	0.34
	PR	0.27	0.40	0.32		0.00	0.00	0.00		0.00	0.00	0.00	
	AV	0.33	0.27	0.30		0.26	0.73	0.38		0.00	0.00	0.00	
	OU	0.37	0.44	0.40		0.33	0.19	0.24		0.42	0.81	0.55	
Transformer	AR	0.78	1.00	0.88	0.59	0.54	1.00	0.70	0.57	0.56	0.71	0.63	0.55
	PR	0.57	0.40	0.47		1.00	0.60	0.75		0.80	0.60	0.71	
	AV	0.41	0.64	0.50		0.40	0.18	0.25		0.00	0.00	0.00	
	OU	0.73	0.50	0.59		0.50	0.62	0.56		0.46	0.81	0.59	
SVM	AR	1.00	0.71	0.83	**0.68**	1.00	0.86	0.92	**0.59**	0.54	0.78	0.64	**0.59**
	PR	0.88	0.78	0.82		1.00	0.33	0.50		1.00	0.20	0.33	
	AV	0.41	0.88	0.56		0.38	0.38	0.38		0.67	0.50	0.57	
	OU	0.67	0.46	0.55		0.38	0.62	0.47		0.57	0.76	0.65	

intervention strategies.

Moreover, identifying students at risk (AR) or prone to risk (PR) early is crucial for timely support. The 2-week model, when fully contextualized, exhibits a remarkable recall rate of 100% for the AR group. As more data becomes available, the 4-week model maintains this 100% recall for the AR group and also achieves an 80% recall for the PR group, both of which are essential for early intervention efficacy. Expanding the data window to 8 weeks further enhances the model's accuracy to 89%, underlining the benefits of full contextualization in improving early detection and intervention outcomes.

Partial Contextualization was explored in two variations: one combining cognitive and non-cognitive features (C + NC) and the other cognitive and background features (C + B). The C + NC configuration demonstrated moderate success, with overall accuracy ranging from 68% to 82%, indicating a somewhat effective use of student information minus the background context. In contrast, the C + B setup, omitting non-cognitive traits, showed a slight decrease in performance, particularly for the 2-week predictions, where accuracy dropped to 64%. These outcomes highlight the nuanced contribution of non-cognitive factors in short-term risk assessment.

No Contextualization (C alone) presented the **most significant drop in performance**, with accuracy falling to 52% for the 2-week predictions. This stark decrease underscores the critical role of contextualization in enhancing the predictive power of the model.

In addressing RQ1, the evaluation of the FLAN T5 Base model also underscores the importance of academic trajectory data contextualization (see Figure 2(a)). When fine-tuned with a comprehensive set of features (C + NC + B), it demonstrates a clear advantage, achieving accuracies of 86%, 84%, and 68% across 8-week, 4-week, and 2-week forecasts, respectively. This trend highlights the efficacy of full contextualization in enhancing model performance, despite a slight performance dip compared to the larger model variant, affirming the significance of a rich feature set for improved predictive accuracy.

The investigation with the FLAN T5 Small model further supports the value of contextualization (see Figure 2(b)), achieving peak accuracies of 82%, 75%, and 64% across the same timeframes with full feature integration. Despite facing challenges in short-term risk prediction, the Small model's performance emphasizes the critical role of a comprehensive feature blend in maintaining predictive accuracy, even with constrained computational resources. These findings collectively validate that full contextualization substantially benefits the forecasting capabilities of pre-trained LMs across different model sizes.

[RQ2]: ***How does natural language text generation compare to numeric feature-based models in forecasting early academic performance, using only cognitive features?*** Our analysis contrasts the efficacy of three varying-capacity LMs against four numeric feature-based baseline models, focusing solely on the cognitive features of our dataset. As illustrated in Figure 3 for datasets spanning 8-week, 4-week, and 2-week intervals, the results demonstrate distinct performance dynamics. In the 4-week and 8-week forecasts, LMs consistently outperform the numeric baseline models. Yet, in the initial 2-week forecast, numeric models, specifically the SVM and Transformer, with accuracies of 59% and 55% respectively, outdo the large LM, which records a 52% accuracy. Remarkably, the SVM's performance plateaus at 59% accuracy for the 4-week datasets, in contrast to the large LM, which notably enhances its accuracy to over 70% consistently across the 4-week duration. Detailed comparisons of baseline model performances are provided in Table 2.

[RQ3]: ***What impact does the capacity of pre-trained LMs (i.e., the number of parameters) have on forecasting accuracy?*** Analyzing the test accuracies among the three differently sized LMs (refer to Table 1, Figures 2 and 3) reveals a clear trend: larger models demonstrate enhanced forecasting capabilities. Notably, even after implementing full contextualization, the recall for the at-risk group in the smaller and medium-sized models stands at 86%, while the large model achieves a recall of 100%. This pattern strongly indicates that **achieving optimal early forecasting through the contextualization of LMs is more effective with the deployment of large language models (LLMs)**.

4 Conclusion

In this paper, we ventured into the realm of leveraging modern AI, particularly deep learning and transfer learning methodologies, to tackle the critical challenge of early performance forecasting in the educational sector. Our investigation centered on the innovative use of Transformer-based pre-trained LMs for predicting undergraduate STEM course outcomes, marking a significant departure from traditional numeric feature-based models. By integrating a novel transfer learning approach tailored for small-domain data within STEM education, we aimed to overcome the limitations posed by sparse training datasets, a common hurdle in the educational domain.

Our methodology hinged on the contextualization of academic trajectory data, incorporating a rich tapestry of both cognitive and non-cognitive factors. Through this multi-dimensional approach, we enhanced the LMs' capacity to understand and predict academic performance, achieving a notable improvement in forecasting accuracy. Specifically, we demonstrated that:

- Contextualizing academic trajectory data significantly enhances the transfer learning process from pre-trained LMs, as evidenced by our responses to [RQ1].
- Compared to numeric feature-based models, our natural language text generation approach shows superior performance in early academic forecasting, addressing [RQ2].
- The capacity of pre-trained LMs, in terms of their number of parameters, plays a crucial

role in forecasting accuracy, with larger models outperforming their smaller counterparts, as explored in [RQ3].

These insights underscore the transformative potential of AI-driven tools in proactively identifying and supporting students at risk, thereby enhancing educational outcomes. By leveraging the vast knowledge encapsulated within LMs and enriching it with detailed contextual data across demographic, socioeconomic, academic, and emotional engagement dimensions, we not only tailored the pre-trained LMs to our specific task but also enriched the predictive model with a comprehensive understanding of students' academic journeys.

Looking ahead, our work opens the door to future research in several key areas. Integrating more detailed contextual signals such as real-time academic engagement and behavioral data could enhance LM predictive accuracy, leveraging advances in natural language processing and sentiment analysis to understand students' emotional and cognitive states better. Expanding our approach to a wider range of educational contexts and disciplines would help validate its scalability and adaptability. Additionally, exploring continual learning techniques for LMs might illuminate how to improve forecasting systems' accuracy and reliability over time without extensive retraining. Addressing the ethical and privacy concerns inherent in using detailed student data is also crucial, necessitating robust data governance and ethical AI frameworks to protect students' rights and ensure equitable benefits.

5 Related Work

In advancing educational forecasting, we introduce a distinct approach by applying transfer learning from pre-trained LMs to contextualized time-series data of academic trajectories. This dataset uniquely incorporates both cognitive and non-cognitive features, enriching the forecasting model with a detailed temporal perspective.

Research in time-series forecasting with pre-trained LMs splits into two main streams: data-centric and model-centric approaches (Sun et al., 2023). **Data-centric** methods focus on transforming time-series data into formats amenable to LMs, employing innovative embedding techniques to match time-series data with the textual embedding space of LMs. These techniques range from embedding alignment and augmentation (Sun et al., 2023) to two-stage fine-tuning (Chang et al., 2023) and zero-shot preprocessing for numerical data (Gruver et al., 2023). **Model-centric** strategies, on the other hand, adapt pre-trained LMs specifically for time-series forecasting. This involves fine-tuning certain LM components while introducing time series-specific modifications such as decomposition and soft prompts (Cao et al., 2023), aiming to formulate forecasting as a question-answering task (Xue and Salim, 2023), and prompt-tuning with few-shot learning (prompt engineering) (Liu et al., 2023c).

Our work diverges by leveraging a model-centric approach tailored to the contextual data of academic paths, utilizing discrete prompts. This novel strategy emphasizes the importance of transfer learning from pre-trained LMs to enrich forecasting with a deep, context-aware analysis, setting our research apart in the field of educational forecasting.

6 Limitations

Our study has made important progress in showing how contextualized language models (LMs) can predict early academic performance. Yet, we must acknowledge some limitations that define our research's scope and point towards future research directions.

Data Scope and Diversity: The primary focus of our research on undergraduate STEM courses may circumscribe the applicability of our findings across different academic disciplines and educational levels. The distinct cognitive and engagement challenges inherent to non-STEM subjects underscore the need for subsequent studies aimed at adapting and validating our methodology in a wider educational context.

Model Size and Computational Resources: The deployment of LMs brings to the fore the exigencies of computational resources. The high computational overhead required for the training and operational deployment of these models might preclude their adoption in institutions with limited technological infrastructure, potentially curtailing the broad-scale application of our approach in varied educational settings.

Ethical and Privacy Concerns: Leveraging detailed personal and contextual data of students necessitates a careful navigation of ethical and privacy considerations. While our study has endeavored to adhere to these imperatives scrupulously, the expansive use of similar methodologies demands a rigorous commitment to data protection standards and ethical practices to mitigate the risk of infringing upon student privacy.

Temporal Dynamics: Our forecasting approach captures a static slice of contextual data, possibly overlooking the dynamic nature of student engagement and performance, which are subject to change over the academic term. The challenge of incorporating continuous data updates into LMs without necessitating extensive retraining poses a significant question for future research.

Interpretability and Explainability: The opaque nature of LMs, as with many deep learning models, presents a barrier to interpretability and explainability. To engender trust among educational practitioners and stakeholders, it is imperative to develop methodologies that elucidate the rationales behind model predictions in a comprehensible manner.

Bias and Fairness: The risk of propagating biases through pre-trained LMs, a reflection of their training datasets, is a critical concern. These biases have the potential to skew forecasting accuracy and fairness, impacting various student demographics disparately. Vigilance to prevent the reinforcement of existing educational disparities is essential.

Computational Limitations: Our investigation's scope was notably constrained by the limited memory capacity of available GPUs. This limitation thwarted our ability to fully leverage the spectrum of distal and proximal non-cognitive features, employ rich and expressive instructional prompts, and utilize LMs with ≥ 1 billion parameters. Overcoming these computational hurdles is crucial for unlocking the full potential of LLMs in educational forecasting.

These limitations underscore the imperative for continued research to surmount these hurdles. Future endeavors should focus on broadening the inclusivity, ethical integrity, and scalability of AI-driven educational interventions, ensuring they serve as equitable and effective support mechanisms across the diverse landscape of learning environments.

Acknowledgments

This research was supported by grants from the U.S. National Science Foundation (NSF DUE 2142558), the U.S. National Institutes of Health (NIH NIGMS P20GM130461 and NIH NIAAA R21AA029231), and the Rural Drug Addiction Research Center at the University of Nebraska-Lincoln. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Science Foundation, National Institutes of Health, or the University of Nebraska.

References

Daniel A Adler, Dror Ben-Zeev, Vincent W-S Tseng, John M Kane, Rachel Brian, Andrew T Campbell, Marta Hauser, Emily A Scherer, and Tanzeem Choudhury. 2020. Predicting Early Warning Signs of Psychotic Relapse From Passive Sensing Data: An Approach Using Encoder-Decoder Neural Networks. *JMIR mHealth and uHealth*, 8(8):e19962.

Daniel A. Adler, Fei Wang, David C. Mohr, and Tanzeem Choudhury. 2022. Machine learning for passive mental health symptom prediction: Generalization across different longitudinal mobile sensing studies. *PLOS ONE*, 17(4):e0266516. Publisher: Public Library of Science.

Kimberly E. Arnold and Matthew D. Pistilli. 2012. Course Signals at Purdue: Using Learning Analytics to Increase Student Success. *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, pages 267–270.

A. Bandura. 2001. Social cognitive theory of mass communication. *Media Psychology*, 3:265–299.

Kush Bhatia, Avanika Narayan, Christopher De Sa, and Christopher Ré. 2023. TART: A plug-and-play Transformer module for task-agnostic reasoning. ArXiv:2306.07536 [cs].

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual Workshop on Computational Learning Theory (COLT'92)*, pages 144–152, Pittsburgh, PA, USA. ACM Press.

Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. 2023. Tempo: Prompt-based generative pre-trained transformer for time series forecasting.

Ching Chang, Wen-Chih Peng, and Tien-Fu Chen. 2023. Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms.

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 Sankaranarayana Pillai, 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. ArXiv:2204.02311 [cs].

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. ArXiv:2210.11416 [cs].

Amanda C. Collins, Damien Lekkas, Matthew David Nemesure, Tess Z. Griffin, George Price, Arvind Pillai, Subigya Nepal, Michael V. Heinz, Andrew T. Campbell, and Nicholas C. Jacobson. 2023. Semantic signals in self-reference: The detection and prediction of depressive symptoms from the daily diary entries of a sample with major depressive disorder.

Nello Cristianini and John Shawe-Taylor. 1999. *An Introduction to Support Vector Machines: And Other Kernel-Based Learning Methods*. Cambridge University Press, USA.

BJ Fogg. 2009. A behavior model for persuasive design. In *Proceedings of the 4th International Conference on Persuasive Technology*, Persuasive '09, pages 1–7, New York, NY, USA. Association for Computing Machinery.

Jennifer Fredricks. 2014. *Eight Myths of Student Disengagement: Creating Classrooms of Deep Learning*. Corwin Press, Thousand Oaks, California.

Nathan Greenstein, Grant Crider-Phillips, Claire Matese, and Sung-Woo Cho. 2021. Predicting Student Outcomes to Drive Proactive Support: An Exploration of Machine Learning to Advance Student Equity & Success. Technical report, University of Oregon.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. 2023. Large language models are zero-shot time series forecasters.

Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. 2017. Learning from class-imbalanced data: Review of methods and applications. *Expert systems with applications*, 73:220–239.

Mohammad Hasan and Bilal Khan. 2023. A Trajectory-Clustering Framework for Assessing AI-Based Adaptive Interventions in Undergraduate STEM Learning American Society for Engineering Education.

Mohammad Rashedul Hasan and Mohamed Aly. 2019. Get More From Less: A Hybrid Machine Learning Framework for Improving Early Predictions in STEM Education. In *The 6th Annual Conf. on Computational Science and Computational Intelligence, CSCI 2019*. Event-place: Las Vegas, Nevada.

Julio Noe Hernandez, Jesús Ariel Carrasco-Ochoa, and José Francisco Martínez Trinidad. 2013. An empirical study of oversampling and undersampling for instance selection methods on imbalance datasets. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications - 18th Iberoamerican Congress, CIARP 2013, Havana, Cuba, November 20-23, 2013, Proceedings, Part I*, volume 8258 of *Lecture Notes in Computer Science*, pages 262–269. Springer.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Adam G. Horwitz, Shane D. Kentopp, Jennifer Cleary, Katherine Ross, Zhenke Wu, Srijan Sen, and Ewa K. Czyz. 2022. Using machine learning with intensive longitudinal data to predict depression and suicidal ideation among medical interns over time. *Psychological Medicine*, pages 1–8.

M. Jahangiri, A. Kazemnejad, K. S. Goldfeld, M. S. Daneshpour, S. Mostafaei, D. Khalili, M. R. Moghadas, and M. Akbarzadeh. 2023. A wide range of missing imputation approaches in longitudinal data: a simulation study and real data analysis. *BMC Med Res Methodol*, 23(1):161.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

John M. Lachin. 2016. Fallacies of last observation carried forward analyses. *Clinical trials*, 13(2):161–168.

Bohan Li, Yutai Hou, and Wanxiang Che. 2022. Data augmentation approaches in natural language processing: A survey. *AI Open*, 3:71–90.

Xiang Li, Xinning Zhu, Xiaoying Zhu, Yang Ji, and Xiaosheng Tang. 2020. Student Academic Performance Prediction Using Deep Multi-source Behavior Sequential Network. In *Advances in Knowledge Discovery and Data Mining*, Lecture Notes in Computer Science, pages 567–579, Cham. Springer International Publishing.

Haihong Liu, Xiaolei Zhang, Haining Liu, and Sheau Tsuey Chong. 2023a. Using Machine Learning to Predict Cognitive Impairment Among Middle-Aged and Older Chinese: A Longitudinal Study. *International Journal of Public Health*, 68:1605322.

Lydia T. Liu, Serena Wang, Tolani Britton, and Rediet Abebe. 2023b. Reimagining the machine learning life cycle to improve educational outcomes of students. *Proceedings of the National Academy of Sciences*, 120(9):e2204781120. Publisher: Proceedings of the National Academy of Sciences.

Xian Liu. 2016. Methods for handling missing data. In Xian Liu, editor, *Methods and Applications of Longitudinal Data Analysis*, chapter 14, pages 441–473. Academic Press.

Xin Liu, Daniel McDuff, Geza Kovacs, Isaac Galatzer-Levy, Jacob Sunshine, Jiening Zhan, Ming-Zher Poh, Shun Liao, Paolo Di Achille, and Shwetak Patel. 2023c. Large language models are few-shot health learners.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Abdullah Mamun, Krista S. Leonard, Matthew P. Buman, and Hassan Ghasemzadeh. 2022. Multimodal Time-Series Activity Forecasting for Adaptive Lifestyle Intervention Design. In *2022 IEEE-EMBS International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pages 1–4. ISSN: 2376-8894.

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):140:5485–140:5551.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How Much Knowledge Can You Pack Into the Parameters of a Language Model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Chenxi Sun, Yaliang Li, Hongyan Li, and Shenda Hong. 2023. Test: Text prototype aligned embedding to activate llm's ability for time series.

Maria Tsiakmaki, Georgios Kostopoulos, Sotiris Kotsiantis, and Omiros Ragos. 2020. Transfer learning from deep neural networks for predicting student performance. *Applied Sciences*, 10(6).

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:1706.03762 [cs].

Rui Wang, Fanglin Chen, Zhenyu Chen, Tianxing Li, Gabriella Harari, Stefanie Tignor, Xia Zhou, Dror Ben-Zeev, and Andrew T. Campbell. 2014. StudentLife: assessing mental health, academic performance and behavioral trends of college students using smartphones. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, pages 3–14, New York, NY, USA. Association for Computing Machinery.

Rui Wang, Peilin Hao, Xia Zhou, Andrew T. Campbell, and Gabriella Harari. 2016. SmartGPA: How Smartphones Can Assess and Predict Academic Performance of College Students. *GetMobile: Mobile Computing and Communications*, 19(4):13–17.

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. ArXiv:2201.11903 [cs].

Weiqi Xu and Fan Ouyang. 2022. The application of AI technologies in STEM education: a systematic review from 2011 to 2021. *International Journal of STEM Education*, 9(1):59.

Xuhai Xu, Xin Liu, Han Zhang, Weichen Wang, Subigya Nepal, Yasaman Sefidgar, Woosuk Seo, Kevin S. Kuehn, Jeremy F. Huckins, Margaret E. Morris, Paula S. Nurius, Eve A. Riskin, Shwetak Patel, Tim Althoff, Andrew Campbell, Anind K. Dey, and Jennifer Mankoff. 2023. GLOBEM: Cross-Dataset Generalization of Longitudinal Human Behavior Modeling. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(4):190:1–190:34.

Hao Xue and Flora D. Salim. 2023. PromptCast: A New Prompt-based Learning Paradigm for Time Series Forecasting. ArXiv:2210.08964 [cs, math, stat].

Juan Zhao, QiPing Feng, Patrick Wu, Roxana A. Lupu, Russell A. Wilke, Quinn S. Wells, Joshua C. Denny, and Wei-Qi Wei. 2019. Learning from Longitudinal Data in Electronic Health Record and Genetic Data to Improve Cardiovascular Event Prediction. *Scientific Reports*, 9:717.

Can GPT-4 do L2 analytic assessment?

Stefano Bannò, Hari K. Vydana, Kate M. Knill, Mark J. F. Gales
ALTA Institute, Department of Engineering, University of Cambridge (UK)
{sb2549,hkv21,kmk1001,mjfg100}@cam.ac.uk

Abstract

Automated essay scoring (AES) to evaluate second language (L2) proficiency has been a firmly established technology used in educational contexts for decades. Although holistic scoring has seen advancements in AES that match or even exceed human performance, analytic scoring still encounters issues as it inherits flaws and shortcomings from the human scoring process. The recent introduction of large language models presents new opportunities for automating the evaluation of specific aspects of L2 writing proficiency. In this paper, we perform a series of experiments using GPT-4 in a zero-shot fashion on a publicly available dataset annotated with holistic scores based on the Common European Framework of Reference and aim to extract detailed information about their underlying analytic components. We observe significant correlations between the automatically predicted analytic scores and multiple features associated with the individual proficiency components.

1 Introduction

Automated essay scoring (AES) of second language (L2) proficiency is a well-established technology in educational settings, involving the automatic scoring and evaluation of learners' written productions through computer programs (Shermis and Burstein, 2003).

Originating in the 1960s, the roots of AES can be traced back to the development of Project Essay Grade (PEG) (Page, 1966, 1968), an automatic system which evaluated writing skills based only on proxy traits: hand-written texts had to be manually entered into a computer, and a scoring algorithm then quantified superficial linguistic features, such as essay length, average word length, count of punctuation, count of pronouns and prepositions, etc. Across the following decades, as natural language processing (NLP) technologies have advanced and increased their power (Landauer, 2003), the field of AES has expanded and improved, and more significant studies have been conducted from the 1990s and early 2000s. The most widely known automated scoring systems for essays include the e-rater®, developed by Educational Testing Service (Burstein, 2002; Attali and Burstein, 2006), IntelliMetric™ by Vantage Learning (Rudner et al., 2006), and the Intelligent Essay Assessor™, built at Pearson Knowledge Technologies (Landauer et al., 2002).

In recent years, deep neural network (DNN) approaches have brought significant improvements (Alikaniotis et al., 2016), and especially the advent of transformer-based architectures (Vaswani et al., 2017), such as BERT (Devlin et al., 2019) which took the world of NLP and, consequently, AES by storm, outperforming classic feature-based systems (Rodriguez et al., 2019). Yet, the most recent breakthrough has been brought by large language models (LLMs), such as the GPT models (Brown et al., 2020; OpenAI, 2023), which might revolutionise the world of AES, not only from the NLP experts' and language testers' perspective, but also considering the users' point of view due to GPT's extremely accessible and intuitive interface. In the context of L2 writing assessment, previous studies have employed GPT-3.5 (Mizumoto and Eguchi, 2023) and GPT-4 (Yancey et al., 2023), obtaining promising results.

Although LLMs have been employed for holistic scoring (i.e., assessing the overall quality of a composition as a whole, considering various aspects such as vocabulary, grammar, coherence, etc. altogether), to the best of our knowledge, so far they have not been investigated for the task of analytic scoring (i.e., breaking down a composition into specific components or criteria and assigning separate scores or ratings to each component).[1] Offering L2

[1] Naismith et al. (2023) investigated the use of GPT-4 on a proprietary dataset annotated with specific scores targeting coherence only.

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 149–164
June 20, 2024 ©2024 Association for Computational Linguistics

learners specific analytic proficiency scores is crucial for delivering insightful and effective feedback, emphasising both their strengths and weaknesses to facilitate improvement.

For holistic scoring, previous works have shown that state-of-the-art automatic techniques can reach near-human results (Alikaniotis et al., 2016; Taghipour and Ng, 2016) or even outperform them (Rodriguez et al., 2019). This is, at least in part, ascribable to the fact that holistic scores are generally easier to obtain for human evaluators (see Section 2). Conversely, assessing analytic aspects of language proficiency is generally considered to be more difficult, time-consuming, and cognitively demanding for human evaluators, and, as a result, "noisy" ground truth scores are harder to learn and predict for automatic systems (see Section 2).

Starting from these premises, in this paper, we conduct a series of exploratory experiments on a publicly available dataset annotated with holistic scores according to the Common European Framework of Reference (CEFR) (Council of Europe, 2001, 2020) using GPT-4 in a zero-shot fashion, and aim to extract specific information about their underlying analytic components. Although ground truth analytic scores are not available, we find significant correlations between the analytic scores predicted by the model and several features related to the analytic scores.

2 Holistic versus analytic scoring

2.1 Human assessment

Holistic and analytic approaches to assessing L2 proficiency are commonly utilised, differing in scoring methods, underlying assumptions, and practical application. While holistic assessment consists of assigning a single overall numerical score to a specific performance based on a singular set of rating criteria, analytic assessment involves providing various sub-scores to the performance based on multiple sets of criteria. As a result, there are conceptual differences between the two approaches (Barkaoui, 2011). Holistic assessment typically assumes that the construct being evaluated is a unitary entity and can be represented on a single scale. While this approach acknowledges that the construct may consist of various elements, it implies that development across various aspects of proficiency is uniform. Conversely, analytic assessment views the construct as multi-dimensional and advocates for a multi-faceted assessment, recognising that development across various aspects may be irregular. For instance, the levels of the CEFR are structured according to 'can-do' descriptors of language proficiency outcomes and expect evaluators to grade proficiency by means of holistic assessments. Nonetheless, the CEFR levels do have a modularisable structure with multiple underlying components (e.g., vocabulary range, vocabulary control, grammatical accuracy, etc.), acknowledging that a learner may be more proficient in certain aspects than others (Council of Europe, 2001, 2020).

When we consider assessment strictly from a human perspective, holistic assessment is considered highly practical as it is more time-efficient per se and in relation to rater training (White, 1984), less cognitively demanding (Xi, 2007), and generally has a higher inter-annotator agreement (Weigle, 2002) than analytic assessment. On the other hand, holistic scoring may suffer from lack of clarity regarding how different aspects are prioritised, which may vary among evaluators (Weigle, 2002; Xi, 2007), the risk that evaluators might primarily concentrate on candidates' strengths rather than their weaknesses (Bacha, 2001), and the potentially erroneous assumption that various aspects of proficiency develop uniformly over time (Kroll, 1990).

Analytic assessment allows for a more detailed and systematic evaluation and is supposed to provide much more detailed feedback to L2 learners, by highlighting their fortes and their weaknesses (Hamp-Lyons, 1995) in addition to enhancing scoring validity. However, it is not a panacea. Analytic scores may be psychometrically redundant (Lee et al., 2009) due to a halo effect (Engelhard, 1994), whereby raters fail to distinguish between different aspects of learners' performances but assess all or some of them with similar scores. For example, when assessing grammatical accuracy, raters might be influenced by the score previously assigned to vocabulary range. On top of this, raters might confuse analytic criteria in the phase of assessment due to high cognitive load (Underhill, 1987; Cai, 2015) or, more simply, to indefiniteness of the analytic criteria (Douglas and Smith, 1997). The difficulty in providing analytic scores — especially for a large number of written productions — is evident in the total absence of publicly available L2 English learner datasets annotated in this way[2] and the fact that the primary emphasis in AES

[2] To the best of our knowledge, the only formerly publicly

research has been on holistic scoring.

2.2 Automatic assessment

The introduction of automatic assessment techniques — and especially their recent advancements — have started to change the game. For holistic scoring, DNN-based systems reached near-human performances (Alikaniotis et al., 2016; Taghipour and Ng, 2016), and the application of transformers-based architectures even beat human inter-annotator agreement (Rodriguez et al., 2019). However, a notorious problem lies in the impossibility to enter the black box of neural scoring models, and this poses a challenge for explainability and interpretability of the machine-generated holistic scores. Even more so, it is important to explore the ability of automatic models to evaluate specific aspects of language proficiency through analytic scoring: if it is not possible to decompose the holistic assessment process by peeking inside the black box, it may be possible to reconstruct holistic scores starting from their analytic components (with the caveat that we should keep in mind the potential unreliability of human analytic scores, as discussed above). In this regard, automatic systems have been found to be generally better at evaluating specific linguistic phenomena, whilst humans tend to focus on more general aspects of proficiency. For example, Enright and Quinlan (2010) suggested that human raters might achieve higher results when assessing ideas, content, and organisation, whereas automatic systems might have better performances when evaluating microfeatures at the grammatical, syntactic, lexical, and discourse levels. It should be noted, however, that these limitations attributed to automatic systems may no longer necessarily be true in light of the recent advancements involving neural systems, which can be used quite effectively also to assess higher-level aspects of proficiency. For example, previous studies have focused on specific traits of written productions, such as organisation, content, word choice, sentence fluency, narrativity, etc. (Hussein et al., 2020; Mathias and Bhattacharyya, 2020; Ridley et al., 2021), but they have used the ASAP dataset, which is problematic for reproducibility and only features essays written by L1 English speakers (see note 2). For L2 speaking assessment, the initial study by Bannò et al. (2022) investigated the use of multiple different graders, each of which focused on a different set of features related to a specific proficiency aspect.

The introduction of LLMs could be a further game-changer, considering their outstanding results in a broad range of tasks.

To sum up, given that:

- holistic scores are generally easier to obtain both from human and automatic graders and generally have a higher inter-annotator agreement, hence higher reliability;
- analytic scores are difficult to obtain and might not always be sufficiently reliable;
- more often than not, L2 learner datasets are annotated with holistic scores only;
- LLMs have been proven to be extremely powerful tools in many NLP tasks;

we pose the following research question:

> is it possible to extract information about analytic aspects from L2 learner essays and their assigned holistic scores using GPT-4?

Figure 1 shows the pipeline adopted in this study, which will be illustrated in detail in Section 4.

3 Data

3.1 Write & Improve

Write & Improve (W&I) is an online platform where L2 learners of English can practise their writing skills (Yannakoudakis et al., 2018). Users can submit their compositions in response to different prompts, and the W&I automatic system provides assessment and feedback. Some of these compositions have been manually annotated with CEFR levels and grammatical error corrections since 2014, resulting in a corpus of 3,300 texts, partitioned into a training set of 3,000 and a validation set of 300 essays.[3] The proficiency scale ranges from A1 to C2 but also has intermediate levels, resulting in 12 levels, that we arranged on a scale from 1 to 6.5, where 1 is A1, 1.5 is A1+, 2 is A2, 2.5 is A2+, etc., as shown in Table 5 (see Appendix D).

available dataset annotated with analytic scores is the ASAP dataset (kaggle.com/c/asap-aes/data), but the test data are no longer available for evaluation and comparison with previous work. Furthermore and most importantly, it contains essays written by L1 English speakers.

[3] The dataset can be downloaded from this link: huggingface.co/datasets/wi_locness.

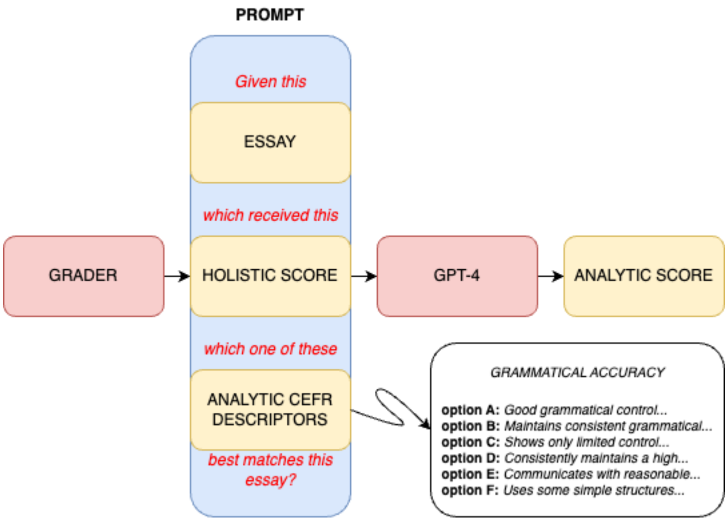

PROMPT
Given this
ESSAY
which received this
GRADER
HOLISTIC SCORE
GPT-4
ANALYTIC SCORE
which one of these
ANALYTIC CEFR DESCRIPTORS
best matches this essay?
GRAMMATICAL ACCURACY
option A: Good grammatical control...
option B: Maintains consistent grammatical...
option C: Shows only limited control...
option D: Consistently maintains a high...
option E: Communicates with reasonable...
option F: Uses some simple structures...

Figure 1: The pipeline presented in this study. Grammatical accuracy is only one of the aspects considered.

3.2 EFCAMDAT

Arguably the largest publicly available[4] L2 learner corpus, the second release of EF-Cambridge Open Language Database (EFCAMDAT) (Geertzen et al., 2013; Huang et al., 2017, 2018) comprises 1,180,310 scripts written by 174,743 L2 learners as assignments to Englishtown, an online English language school. The compositions are annotated with a score on a scale from 0 to 100 and a proficiency level from 1 to 16 (mapped to CEFR levels from A1 to C2).[5] In order to align them to the proficiency levels in the W&I dataset, we normalised the scores as described in Table 5 (see Appendix D). For our experiments, we selected a subset of data consisting of 753,508 essays for the training set and 7612 for the validation set, following a similar process to Bannò et al. (2023).

4 Experimental setup

4.1 Longformer-based holistic grader

Following the pipeline illustrated in Figure 1, we start our experiments from training a holistic grader, which consists of a Longformer model (Beltagy et al., 2020) in the version provided by the HuggingFace Transformer Library,[6] a dropout layer, a dense layer of 768 nodes, a dropout layer, another dense layer of 128 nodes, and finally, the output layer. The baseline model (W&I) is trained on the W&I training data and optimised on the W&I validation data using an Adam optimiser (Kingma and Ba, 2014) for 3 epochs with batch size 16, learning rate 1e-6 and mean squared error as loss, but our best-performing model — which is the one we will use in the following steps of our pipeline — is trained on the EFCAMDAT training set and optimised on the validation data from the same dataset for 0.5 epochs with batch size 16 and learning rate 1e-5, and subsequently fine-tuned on the W&I training data and optimised on the W&I validation data for 4 epochs.

To evaluate the holistic grader performance, we use Pearson's correlation coefficient (PCC), Spearman's rank coefficient (SRC), and root-mean-square error (RMSE).

4.2 GPT-4-based analytic graders

Once we obtain the holistic scores from the Longformer-based model, we move on to feeding them into GPT-4 (*"gpt-4-1106-preview"*) to extract analytic scores. Specifically, the analytic scores are related to 9 proficiency aspects as described in Council of Europe (2020), reported in Appendix A. Five of them compose the linguistic competence: *general linguistic range, vocabulary range, grammatical accuracy, vocabulary control,* and *orthographic control*; while the remaining four form the pragmatic competence: *flexibility, thematic development, coherence and cohesion,* and *propositional precision*.

We excluded sociolinguistic appropriateness because it is not consistently elicited in the W&I essays, as well as the aspects strictly related to speaking proficiency (i.e., phonological control, turntaking, and fluency) for obvious reasons.

[4] ef-lab.mmll.cam.ac.uk/EFCAMDAT.html

[5] englishlive.ef.com/en/how-it-works/levels-and-certificates/

[6] huggingface.co/allenai/longformer-base-4096

The prompt given to GPT-4 can be found in Appendix C. To exclude potential biases, the holistic scores are fed in their numerical form (i.e., from 1 to 6.5) instead of the original CEFR notation (i.e., from A1 to C2+), and the analytic CEFR descriptors are provided in random order and, obviously, without any reference to the CEFR levels. For completeness, we also try this experiment without giving GPT-4 the holistic score.

At the end of the process, the option selected by GPT-4 is mapped back to its respective CEFR level.

4.3 Explanation of the features

As mentioned in Section 1, the W&I dataset does not include analytic scores, but we find significant correlations with relevant features extracted from the essays (see Tables 3 and 4).

%gram. refers to the grammatical error rate, which is the number of grammatical error edits divided by the number of words in the essay. These edits are extracted by feeding the original and corrected versions of the W&I essays into the ERRor ANnotation Toolkit (ERRANT) (Bryant et al., 2017).

#dif.wds. is the number of unique difficult words extracted with `textstat`.[7]

#unq.wds. refers to the number of unique words.

%l.d.t. is the percentage of text types that are content words obtained using TAACO (Tool for the Automatic Analysis of Text Cohesion) 2.0 (Crossley et al., 2019).

#unq.n.chunks refers to the number of unique noun chunks identified and extracted using spaCy. [8]

#unq.q.m.a. refers to the number of unique qualifiers, modality markers, and ambiguity indicators identified and extracted using spaCy.

fl.-kinc. is the Flesch Kincaid readability score (Kincaid et al., 1975), obtained using `textstat`.

w2v is the average word2vec (Mikolov et al., 2013) similarity score between all adjacent paragraphs, extracted with TAACO 2.0.[9]

av.s.ln. is the average sentence length.

The correlations between these features and the analytic scores are evaluated using SRC since we do not necessarily expect a linear correlation between the two. For example, it is well-known that certain grammatical errors are absent or rare in the A1 level, increase after B1, and then decline again by C2 (Hawkins and Buttery, 2010).

5 Experimental results

5.1 Holistic scoring

Table 1 shows the results of the Longformer-based holistic graders on the W&I validation set in terms of PCC, SRC, and RMSE. The model pre-trained on EFCAMDAT and fine-tuned on the W&I training set outperforms the baseline across all metrics as expected. These results should confirm that holistic grading is a relatively easy task and, since the training data are fully publicly available, potentially within everyone's reach.

Model	PCC	SRC	RMSE
W&I	0.707	0.772	1.267
EFC+W&I	0.866	0.874	0.786

Table 1: Holistic scoring results on W&I validation set.

5.2 Holistic score reconstruction

Once we obtain the holistic scores from the Longformer-based grader, we are ready to feed them into GPT-4. However, before moving on to the analysis of the individual analytic scores, we first calculate the correlation between the average of the predicted analytic scores — when providing GPT-4 with the holistic scores from the ground truth (GT) or the Longformer-based grader (EFC+W&I), or with no holistic score (-) — and the holistic scores, both the ground truth (GT) and the scores automatically predicted by the Longformer-based grader (EFC+W&I), as shown in Table 2.

GPT-4 Prompt	Reference	
Holistic Score	GT	EFC+W&I
GT	0.904	0.874
EFC+W&I	0.828	0.898
-	0.797	0.827

Table 2: SRC correlation between the average of the predicted analytic scores and the holistic scores.

The first result that catches the eye is that GPT-4 reaches a significant correlation of 0.797 when it is not provided with additional information about holistic scores (-), although this does not necessarily mean that all the underlying analytic scores

[7] `github.com/textstat/textstat`

[8] `spacy.io/`

[9] Initially, we also extracted the similarity score using Latent Semantic Analysis (Landauer et al., 1998) and Latent Dirichlet Allocation (Blei et al., 2003), which showed similar figures, but we did not include them due to reasons of space.

are effectively targeting their respective proficiency aspects, as we will discuss in the next section. Secondly, it is interesting to observe that the two sources of holistic score in the prompts (i.e., GT and EFC+W&I) result in the information derived from these scores being used in a non-deterministic fashion, introducing a certain degree of variability.

5.3 Analytic scoring

We can now move on to discussing the results of analytic scoring. Table 3 shows the correlation results in terms of SRC between the predicted analytic scores and several relevant features for each proficiency aspect. Table 4 does the same but giving GPT-4 the ground truth holistic scores instead of the scores predicted by the holistic grader. Particularly in the latter, when focusing on the results highlighted in bold, we can observe a broad trend towards an approximate diagonal which passes through most of the proficiency aspects of the linguistic (Lng.) and pragmatic (Prg.) competences on the y-axis and the relevant features on the x-axis. For completeness, in Table 6 (see Appendix D), we also report the results obtained without giving GPT-4 the holistic score, but the correlations are not as significant as the ones shown in Tables 3 and 4 as the holistic score seems to work as a guide for analytic scoring. Furthermore, as expected, the correlations between each individual predicted analytic score and the holistic scores are significantly lower than the ones reported in Tables 3 and 4. Therefore, our analysis in the following lines will not dwell on these results.

As expected, grammatical error rate (%gram.) shows the highest correlations with the aspects of grammatical accuracy and orthographic control both on Tables 3 and 4.

The number of unique difficult words (#dif.wds.) seems to be a suitable feature to measure vocabulary control, e.g., if we compare the A2 level (i.e., “Can control a narrow repertoire dealing with concrete, everyday needs.”) and the C1 level (i.e., “Uses less common vocabulary idiomatically and appropriately.”), as described in Council of Europe (2020, pp. 132-133) (see Appendix A). Indeed, this feature shows the highest correlation with the score related to vocabulary control.

If we look at the results obtained giving the ground truth holistic scores to GPT-4 shown in Table 4, we can see that the number of unique words (#unq.wds.), the percentage of lexical density types (%l.d.t.), and the number of unique noun chunks (#unq.n.cks.), which are all related to lexicality, have their highest correlation with the two scores related to vocabulary. As expected, the same features have slightly weaker — but still relevant — correlations when we use the automatically predicted holistic scores, as shown in Table 3.

The number of unique qualifiers, modality markers, and ambiguity indicators (#unq.q.m.a.) is supposed to be a measure for propositional precision since, for example, as shown in Appendix A, a C1-level learner “[c]an qualify opinions and statements precisely in relation to degrees of, for example, certainty/uncertainty, belief/doubt, likelihood, etc” and “[c]an make effective use of linguistic modality to signal the strength of a claim, an argument or a position”, and a C2-level learner “[c]an convey finer shades of meaning precisely by using, with reasonable accuracy, a wide range of qualifying devices [...]” and “[c]an give emphasis, differentiate and eliminate ambiguity” (Council of Europe, 2020, p. 141). As can be observed in Table 4, this feature has the second-highest correlation with the propositional precision score and the highest correlation with the score related to vocabulary control, with which it is in fact connected. Similarly to what we observed about the lexical features, the results of the fully-automated pipeline for this feature are less evident, but we can still see a rather high correlation with propositional precision.

Given its emphasis on precision and clarity, we thought that also the Flesch-Kincaid readability score (fl.kinc.) would be a suitable feature to measure these. We found that the highest correlation was exactly with propositional precision followed by vocabulary control on both Tables 3 and 4.

Furthermore, we considered two features for the pragmatic competence, especially in relation to cohesion and coherence. The first one is the average word2vec similarity score between all adjacent paragraphs (w2v), which shows the highest correlations on propositional precision and cohesion and coherence in Table 4. The second is average sentence length (av.s.ln.), which should be an indicator of higher use of subordination and cohesive devices (i.e., longer sentences should generally be more complex). This feature shows similar results, as shown in Table 4. When using the scores provided by the automatic holistic grader, the results on both features are also slightly weaker (see Table 3), as observed already for other features above.

It is rather difficult to provide a precise and exhaustive explanation of the results for the general

		score	%gram.	#dif.wds.	#unq.wds.	%l.d.t.	#unq.n.cks.	#unq.q.m.a.	fl.-kinc.	w2v	av.s.ln.	holistic
Lng.	gen. lin.	0.695	0.584	0.514	0.400	0.493	0.527	0.259	0.258	0.143	0.765
	gramm.	**0.698**	0.505	0.469	0.370	0.423	0.468	0.189	0.265	0.134	0.737
	orth.	**0.718**	0.395	0.317	0.244	0.291	0.350	0.155	0.206	0.073	0.652
	voc. ctrl.	0.652	**0.638**	**0.580**	**0.445**	0.537	**0.600**	**0.263**	0.291	**0.189**	0.779
	voc. rg.	0.651	**0.621**	0.568	0.424	**0.548**	0.576	0.254	**0.339**	0.177	0.749
Prg.	propos.	0.601	0.607	0.545	0.389	0.528	0.568	**0.294**	**0.351**	**0.202**	0.702
	coh.	0.662	**0.621**	**0.574**	0.410	**0.551**	**0.588**	0.248	0.336	0.180	0.774
	flexib.	0.424	0.414	0.390	0.291	0.367	0.412	0.178	0.195	0.125	0.443
	themat.	0.584	0.544	0.527	**0.428**	0.516	0.534	0.203	0.287	0.145	0.650
	holistic	0.732	0.640	0.665	0.451	0.623	0.637	0.178	0.364	0.141	1.000

Table 3: SRC correlation of the GPT-4 predicted scores and relevant linguistic features (**using holistic scores predicted by the Longformer-based grader**). The *holistic* entry refers to the ground-truth holistic scores. In bold the two highest correlations columnwise.

		score	%gram.	#dif.wds.	#unq.wds.	%l.d.t.	#unq.n.cks.	#unq.q.m.a.	fl.-kinc.	w2v	av.s.ln.	holistic
Lng.	gen. lin.	0.726	0.574	0.541	0.414	0.522	0.519	0.197	0.267	0.129	0.814
	gramm.	**0.731**	0.472	0.464	0.363	0.433	0.450	0.100	0.286	0.030	0.791
	orth.	**0.726**	0.436	0.398	0.310	0.354	0.427	0.146	0.203	0.060	0.729
	voc. ctrl.	0.674	**0.640**	**0.621**	**0.453**	**0.591**	**0.624**	**0.243**	0.319	0.179	0.854
	voc. rg.	0.672	**0.624**	**0.582**	**0.452**	**0.563**	0.573	0.218	0.280	0.134	0.816
Prg.	propos.	0.600	**0.624**	0.581	0.417	0.560	**0.593**	**0.261**	**0.353**	**0.190**	0.771
	coh.	0.702	0.555	0.534	0.372	0.511	0.535	0.238	**0.339**	**0.201**	0.827
	flexib.	0.425	0.370	0.368	0.249	0.357	0.368	0.140	0.163	0.104	0.488
	themat.	0.639	0.514	0.504	0.413	0.492	0.483	0.224	0.264	0.179	0.745
	holistic	0.732	0.640	0.665	0.451	0.623	0.637	0.178	0.364	0.141	1.000

Table 4: SRC correlation of the GPT-4 predicted scores and relevant linguistic features (**using ground truth holistic scores**). The *holistic* entry refers to the ground-truth holistic scores. In bold the two highest correlations columnwise.

linguistic range score, which is a broad indicator by definition since it includes elements of grammatical accuracy, syntactic complexity, and vocabulary, and, as a result, shows strong correlations with multiple features. On the other hand, the aspect of flexibility seems to be a little problematic with respect to both the features and the holistic score, probably also due to its "longitudinality", since it seems to be evaluated in relation to previous performances, according to its descriptors (see Appendix A).

Finally, we selected some essays in which there was a large discrepancy between two or more analytic scores, and we evaluated them impressionistically. One example can be found in Appendix B. If we focus on the highest and lowest scores, we notice vocabulary range and orthographic control on one hand, and coherence and cohesion on the other hand. Although quite extreme, this discrepancy makes sense, considering that the learner uses almost no connectors at all and mostly uses coordinating clauses (or even parataxis), but has quite a rich vocabulary and makes no orthographic errors (except for punctuation).

5.4 Statistical tests

Additionally, we explore the relationships among analytic scores using a repeated measures design in order to assess whether there are significant differences among them. While the repeated measures analysis of variance (rANOVA) is a widely known approach for such designs, our data fail to meet the assumptions of sphericity and normality required for its application. Hence, we employ the Friedman test (Friedman, 1937), known as the non-parametric equivalent of rANOVA. This test assesses whether there are significant differences in ranks among multiple paired groups. With a significant p-value obtained, we confirm significant differences among the analytic scores. To determine which scores show significant differences, we conduct post-hoc multiple comparisons using the Ne-

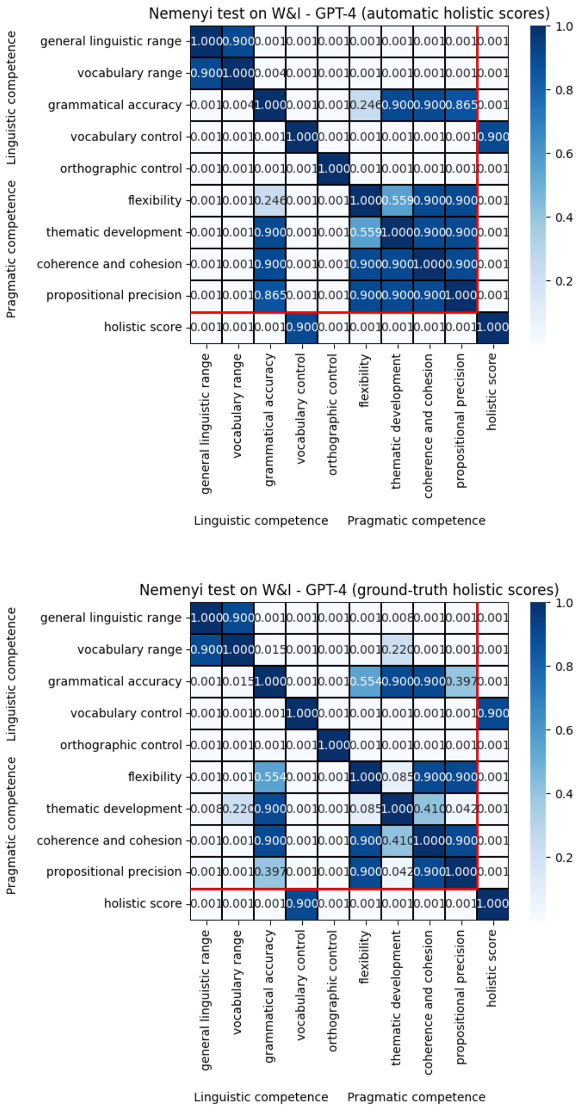


Figure 2: Results of the post-hoc Nemenyi test.

menyi test (Nemenyi, 1963), whose results are reported in Figure 2. The majority of the paired comparisons, even those with the holistic score (except when paired with vocabulary control), show significant differences (i.e., p-value<0.05) both when we provide the ground truth and the automatic holistic scores to GPT-4. In addition to the pairs "general linguistic range - vocabulary range" and "thematic development - vocabulary range", which have some clear overlaps in their descriptors, there seem be non-significant differences over the group of aspects related to the pragmatic competence (i.e., flexibility, thematic development, coherence and cohesion, and propositional precision) and the aspect of grammatical accuracy. While we could expect to see non-significant differences among the aspects related to the pragmatic competence due to their frequent overlaps, the non-significant differences of these with grammatical accuracy might be explained with the fact that not only do its descriptors stress the importance of correctness but, as shown in Appendix A, they also emphasise complexity (e.g., for A1: "Shows only limited control of a few simple grammatical structures [...]"; for B2: "Has a good command of simple language structures and some complex grammatical forms [...]"), which is inherently connected to aspects such as thematic development and coherence and cohesion (Purpura, 2004). In this regard, it is also worth noting that the coherence and cohesion score is the third most correlated with grammatical error rate.

To sum up, under ideal conditions, GPT-4 appears to produce analytic scores that are very reasonably related to the proficiency aspects they are expected to evaluate. The fully-automated pipeline is not always consistent with the ideal system but generates results that are mostly in line with it. This is especially evident for the scores pertaining to grammar and vocabulary.

6 Conclusions

In this paper, we have conducted an initial study on the use of GPT-4 for assessing 9 individual aspects of L2 writing underlying the CEFR proficiency levels in a zero-shot fashion. To do this, we used a holistic grading system on the essays of the W&I validation set and, subsequently, fed them with their respective holistic scores into GPT-4, asking to assess one individual aspect at a time. Although the ground truth analytic scores are not available, we have obtained significant correlations between the predicted analytic scores and various features linked to the componential aspects of the CEFR levels. Beyond its immediate implications for computer-assisted language learning applications, we believe that our exploratory experiments may hold promise as valuable contributions to theoretical studies on construct validity in the broader field of language testing and assessment, given the inclusion of CEFR descriptors in our study.

In order to collect further evidence to support our findings, we plan to deploy this system, use it in educational settings, and evaluate its effectiveness by monitoring learners' progress in relation to each specific aspect of proficiency. Future work will also explore the use of multi-modal systems, such as the one presented in Tang et al. (2023), for assessing L2 speech in a similar fashion.

Limitations

The main limitation of this study is clearly the lack of ground truth analytic scores. The reader should keep in mind, however, that, as mentioned in Section 2, human analytic scoring is often an extremely difficult process, which might not produce completely reliable information. As evidence of this, the absence of publicly available L2 English learner datasets annotated with analytic scores speaks loud and clear and is not only an issue for the objectives of this paper, but for the whole scientific community.

Acknowledgements

This paper reports on research supported by Cambridge University Press & Assessment, a department of The Chancellor, Masters, and Scholars of the University of Cambridge. The authors would like to thank the ALTA Spoken Language Processing Technology Project Team for general discussions and contributions to the evaluation infrastructure.

References

Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. 2016. Automatic text scoring using neural networks. In *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*, pages 715–725. Association for Computational Linguistics (ACL).

Yigal Attali and Jill Burstein. 2006. Automated essay scoring with e-rater® v. 2. *The Journal of Technology, Learning and Assessment*, 4(3).

Nahla Bacha. 2001. Writing evaluation: what can analytic versus holistic essay scoring tell us? *System*, 29(3):371–383.

Stefano Bannò, Bhanu Balusu, Mark J. F. Gales, Kate M. Knill, and Konstantinos Kyriakopoulos. 2022. View-specific assessment of L2 spoken English. In *Proceedings of Interspeech 2022*, pages 4471–4475.

Stefano Bannò, Michela Rais, and Marco Matassoni. 2023. Grammatical Error Correction for L2 Speech Using Publicly Available Data. In *Proc. 9th Workshop on Speech and Language Technology in Education (SLaTE)*, pages 136–140.

Khaled Barkaoui. 2011. Effects of marking method and rater experience on ESL essay scores and rater performance. *Assessment in Education: Principles, Policy & Practice*, 18(3):279–293.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent Dirichlet Allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

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.

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.

Jill Burstein. 2002. The e-rater scoring engine: automated essay scoring with natural language processing. In M. D. Shermis and J. Burstein, editors, *Automated essay scoring: a cross-disciplinary perspective*, pages 113–122. Routledge, New York.

Hongwen Cai. 2015. Weight-Based Classification of Raters and Rater Cognition in an EFL Speaking Test. *Language Assessment Quarterly*, 12(3):262–282.

Council of Europe. 2001. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment*. Cambridge University Press, Cambridge.

Council of Europe. 2020. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment - Companion volume*. Council of Europe, Strasbourg.

Scott A Crossley, Kristopher Kyle, and Mihai Dascalu. 2019. The tool for the automatic analysis of cohesion 2.0: Integrating semantic similarity and text overlap. *Behavior research methods*, 51(1):14–27.

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.

Dan Douglas and Jan Smith. 1997. *Theoretical underpinnings of the Test of Spoken English revision project*. Educational Testing Service Princeton, NJ.

George Engelhard. 1994. Examining Rater Errors in the Assessment of Written Composition with a Many-Faceted Rasch Model. *Journal of Educational Measurement*, 31(2):93–112.

Mary K. Enright and Thomas Quinlan. 2010. Complementing human judgment of essays written by English language learners with e-rater® scoring. *Language Testing*, 3(27):317–334.

Milton Friedman. 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701.

Jeroen Geertzen, Theodora Alexopoulou, and Anna Korhonen. 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*, pages 240–254, Somerville. Cascadilla Proceedings Project.

Liz Hamp-Lyons. 1995. Rating nonnative writing: The trouble with holistic scoring. *TESOL Quarterly*, 29(4):759–762.

John A. Hawkins and Paula Buttery. 2010. Criterial features in learner corpora: Theory and illustrations. *English Profile Journal*, 1:1–23.

Yan Huang, Jeroen Geertzen, Rachel Baker, Anna Korhonen, and Theodora Alexopoulou. 2017. The EF Cambridge Open Language Database (EFCAMDAT): Information for users.

Yan Huang, Akira Murakami, Theodora Alexopoulou, and Anna Korhonen. 2018. Dependency parsing of learner English. *International Journal of Corpus Linguistics*, 23(1):28–54.

Mohamed A. Hussein, Hesham A. Hassan, and Mohammad Nassef. 2020. A trait-based deep learning automated essay scoring system with adaptive feedback. *International Journal of Advanced Computer Science and Applications*, 11(5).

J. Peter Kincaid, Robert P. Fishburne Jr, Richard L. Rogers, and Brad S. Chissom. 1975. Derivation Of New Readability Formulas (Automated Readability Index, Fog Count And Flesch Reading Ease Formula) For Navy Enlisted Personnel. Technical report, Tech. Rep. Naval Technical Training Command - Millington TN Research Branch.

Diederick P. Kingma and Jimmy Ba. 2014. Adam: a method for stochastic optimization. In *International Conference on Learning Representations*.

Barbara Kroll. 1990. *Second Language Writing (Cambridge Applied Linguistics): Research Insights for the Classroom*. Cambridge Applied Linguistics. Cambridge University Press.

Thomas K. Landauer. 2003. Automatic essay assessment. *Assessment in education: principles, policy and practice*, 10(3):295–308.

Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. 1998. An Introduction to Latent Semantic Analysis. *Discourse processes*, 25(2-3):259–284.

Thomas K. Landauer, Darrell Laham, and Peter W. Foltz. 2002. Automated scoring and annotation of essays with the Intelligent Essay Assessor™. In M. D. Shermis and J. Burstein, editors, *Automated essay scoring: a cross-disciplinary perspective*, pages 87–112. Routledge, New York.

Yong-Won Lee, Claudia Gentile, and Robert Kantor. 2009. Toward Automated Multi-trait Scoring of Essays: Investigating Links among Holistic, Analytic, and Text Feature Scores. *Applied Linguistics*, 31(3):391–417.

Sandeep Mathias and Pushpak Bhattacharyya. 2020. Can neural networks automatically score essay traits? In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 85–91, Seattle, WA, USA → Online. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

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.

Ben Naismith, Phoebe Mulcaire, and Jill Burstein. 2023. Automated evaluation of written discourse coherence using GPT-4. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 394–403, Toronto, Canada. Association for Computational Linguistics.

Peter B. Nemenyi. 1963. *Distribution-free Multiple Comparisons*. Ph.D. thesis, Princeton University.

OpenAI. 2023. GPT-4 Technical Report.

Ellis B. Page. 1966. The imminence of... grading essays by computer. *The Phi Delta Kappan*, 47(5):238–243.

Ellis B. Page. 1968. The use of the computer in analyzing student essays. *International Review of Education*, 14(2):210–225.

James E. Purpura. 2004. *Assessing Grammar*. Cambridge Language Assessment. Cambridge University Press.

Robert Ridley, Liang He, Xin-yu Dai, Shujian Huang, and Jiajun Chen. 2021. Automated cross-prompt scoring of essay traits. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13745–13753.

Pedro Uria Rodriguez, Amir Jafari, and Christopher M. Ormerod. 2019. Language models and automated essay scoring.

Lawrence M. Rudner, Veronica Garcia, and Catherine Welch. 2006. An evaluation of intellimetric™essay scoring system. *The Journal of Technology, Learning and Assessment*, 4(4).

Mark D. Shermis and Jill Burstein. 2003. *Automated Essay Scoring: A Cross-disciplinary Perspective*. Routledge, New York.

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.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. 2023. SALMONN: Towards Generic Hearing Abilities for Large Language Models.

Nic Underhill. 1987. *Testing spoken language: A handbook of oral testing techniques*. 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.

Sara Cushing Weigle. 2002. *Assessing Writing*. Cambridge Language Assessment. Cambridge University Press.

Edward M. White. 1984. Holisticism. *College Composition and Communication*, 35(4):400–409.

Xiaoming Xi. 2007. Evaluating analytic scoring for the TOEFL® Academic Speaking Test (TAST) for operational use. *Language Testing*, 24(2):251–286.

Kevin P Yancey, Geoffrey Laflair, Anthony Verardi, and Jill Burstein. 2023. Rating short L2 essays on the CEFR scale with GPT-4. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 576–584.

Helen Yannakoudakis, Øistein E Andersen, Ardeshir Geranpayeh, Ted Briscoe, and Diane Nicholls. 2018. Developing an automated writing placement system for ESL learners. *Applied Measurement in Education*, 31(3):251–267.

A Appendix A

LINGUISTIC COMPETENCE

General linguistic range

A1: Has a very basic range of simple expressions about personal details and needs of a concrete type. Can use some basic structures in one-clause sentences with some omission or reduction of elements.

A2: Has a repertoire of basic language which enables them to deal with everyday situations with predictable content, though they will generally have to compromise the message and search for words/signs. Can produce brief, everyday expressions in order to satisfy simple needs of a concrete type (e.g. personal details, daily routines, wants and needs, requests for information). Can use basic sentence patterns and communicate with memorised phrases, groups of a few words/signs and formulae about themselves and other people, what they do, places, possessions, etc. Has a limited repertoire of short, memorised phrases covering predictable survival situations; frequent breakdowns and misunderstandings occur in non-routine situations.

B1: Has a sufficient range of language to describe unpredictable situations, explain the main points in an idea or problem with reasonable precision and express thoughts on abstract or cultural topics such as music and film. Has enough language to get by, with sufficient vocabulary to express themselves with some hesitation and circumlocutions on topics such as family, hobbies and interests, work, travel and current events, but lexical limitations cause repetition and even difficulty with formulation at times.

B2: Can express themselves clearly without much sign of having to restrict what they want to say. Has a sufficient range of language to be able to give clear descriptions, express viewpoints and develop arguments without much conspicuous searching for words/signs, using some complex sentence forms to do so.

C1: Can use a broad range of complex grammatical structures appropriately and with considerable flexibility. Can select an appropriate formulation from a broad range of language to express themselves clearly, without having to restrict what they want to say.

C2: Can exploit a comprehensive and reliable mastery of a very wide range of language to formulate thoughts precisely, give emphasis, differentiate and eliminate ambiguity. No signs of having to restrict what they want to say.

Vocabulary range

A1: Has a basic vocabulary repertoire of words/signs and phrases related to particular concrete situations.

A2: Has sufficient vocabulary to conduct routine everyday transactions involving familiar situations and topics. Has sufficient vocabulary for the expression of basic communicative needs. Has sufficient vocabulary for coping with simple survival needs.

B1: Has a good range of vocabulary related to familiar topics and everyday situations. Has sufficient vocabulary to express themselves with some circumlocutions on most topics pertinent to their everyday life such as family, hobbies and interests,

work, travel and current events.
B2: Can understand and use the main technical terminology of their field, when discussing their area of specialisation with other specialists. Has a good range of vocabulary for matters connected to their field and most general topics. Can vary formulation to avoid frequent repetition, but lexical gaps can still cause hesitation and circumlocution. Can produce appropriate collocations of many words/signs in most contexts fairly systematically. Can understand and use much of the specialist vocabulary of their field but has problems with specialist terminology outside it.
C1: Has a good command of a broad lexical repertoire allowing gaps to be readily overcome with circumlocutions; little obvious searching for expressions or avoidance strategies. Can select from several vocabulary options in almost all situations by exploiting synonyms of even words/ signs less commonly encountered. Has a good command of common idiomatic expressions and colloquialisms; can play with words/signs fairly well. Can understand and use appropriately the range of technical vocabulary and idiomatic expressions common to their area of specialisation.
C2: Has a good command of a very broad lexical repertoire including idiomatic expressions and colloquialisms; shows awareness of connotative levels of meaning.

Grammatical accuracy

A1: Shows only limited control of a few simple grammatical structures and sentence patterns in a learnt repertoire.
A2: Uses some simple structures correctly, but still systematically makes basic mistakes; nevertheless, it is usually clear what they are trying to say.
B1: Communicates with reasonable accuracy in familiar contexts; generally good control, though with noticeable mother-tongue influence. Errors occur, but it is clear what they are trying to express. Uses reasonably accurately a repertoire of frequently used "routines" and patterns associated with more predictable situations.
B2: Good grammatical control; occasional "slips" or non-systematic errors and minor flaws in sentence structure may still occur, but they are rare and can often be corrected in retrospect. Shows a relatively high degree of grammatical control. Does not make mistakes which lead to misunderstanding. Has a good command of simple language structures and some complex grammatical forms, although they tend to use complex structures rigidly with some inaccuracy.
C1: Consistently maintains a high degree of grammatical accuracy; errors are rare and difficult to spot.
C2: Maintains consistent grammatical control of complex language, even while attention is otherwise engaged (e.g. in forward planning, in monitoring others' reactions).

Vocabulary control

A1: No descriptors available.
A2: Can control a narrow repertoire dealing with concrete, everyday needs.
B1: Shows good control of elementary vocabulary but major errors still occur when expressing more complex thoughts or handling unfamiliar topics and situations. Uses a wide range of simple vocabulary appropriately when discussing familiar topics.
B2: Lexical accuracy is generally high, though some confusion and incorrect word/sign choice does occur without hindering communication.
C1: Uses less common vocabulary idiomatically and appropriately. Occasional minor slips, but no significant vocabulary errors.
C2: Consistently correct and appropriate use of vocabulary.

Orthographic control

A1: Can copy familiar words and short phrases, e.g. simple signs or instructions, names of everyday objects, names of shops, and set phrases used regularly. Can spell their address, nationality and other personal details. Can use basic punctuation (e.g. full stops, question marks).
A2: Can copy short sentences on everyday subjects, e.g. directions on how to get somewhere. Can write with reasonable phonetic accuracy (but not necessarily fully standard spelling) short words that are in their oral vocabulary.
B1: Can produce continuous writing which is generally intelligible throughout. Spelling, punctuation and layout are accurate enough to be followed most of the time.
B2: Can produce clearly intelligible, continuous writing which follows standard layout and paragraphing conventions. Spelling and punctuation are reasonably accurate but may show signs of mother-tongue influence.
C1: Layout, paragraphing and punctuation are consistent and helpful. Spelling is accurate, apart from occasional slips of the pen.

C2: Writing is orthographically free of error.

PRAGMATIC COMPETENCE

Flexibility

A1: No descriptors available.
A2: Can adapt well-rehearsed, memorised, simple phrases to particular circumstances through limited lexical substitution. Can expand learnt phrases through simple recombinations of their elements.
B1: Can adapt their expression to deal with less routine, even difficult, situations. Can exploit a wide range of simple language flexibly to express much of what they want.
B2: Can adjust what they say and the means of expressing it to the situation and the recipient and adopt a level of formality appropriate to the circumstances. Can adjust to the changes of direction, style and emphasis normally found in conversation. Can vary formulation of what they want to say. Can reformulate an idea to emphasise or explain a point.
C1: Can make a positive impact on an intended audience by effectively varying style of expression and sentence length, use of advanced vocabulary and word order. Can modify their expression to express degrees of commitment or hesitation, confidence or uncertainty.
C2: Shows great flexibility in reformulating ideas in differing linguistic forms to give emphasis, differentiate according to the situation, interlocutor, etc. and to eliminate ambiguity.

Thematic development

A1: No descriptors available.
A2: Can tell a story or describe something in a simple list of points. Can give an example of something in a very simple text using "like" or "for example".
B1: Can clearly signal chronological sequence in narrative text. Can develop an argument well enough to be followed without difficulty most of the time. Shows awareness of the conventional structure of the text type concerned when communicating their ideas. Can reasonably fluently relate a straightforward narrative or description as a sequence of points.
B2: Can develop an argument systematically with appropriate highlighting of significant points, and relevant supporting detail. Can present and respond to complex lines of argument convincingly. Can follow the conventional structure of the communicative task concerned when communicating their ideas. Can develop a clear description or narrative, expanding and supporting their main points with relevant supporting detail and examples. Can develop a clear argument, expanding and supporting their points of view at some length with subsidiary points and relevant examples. Can evaluate the advantages and disadvantages of various options. Can clearly signal the difference between fact and opinion.
C1: Can use the conventions of the type of text concerned to hold the target reader's attention and communicate complex ideas. Can give elaborate descriptions and narratives, integrating sub-themes, developing particular points and rounding off with an appropriate conclusion. Can write a suitable introduction and conclusion to a long, complex text. Can expand and support the main points at some length with subsidiary points, reasons and relevant examples.
C2: Can use the conventions of the type of text concerned with sufficient flexibility to communicate complex ideas in an effective way, holding the target reader's attention with ease and fulfilling all communicative purposes.

Propositional precision

A1: Can communicate basic information about personal details and needs of a concrete type in a simple way.
A2: Can communicate what they want to say in a simple and direct exchange of limited information on familiar and routine matters, but in other situations they generally have to compromise the message.
B1: Can explain the main points in an idea or problem with reasonable precision. Can convey simple, straightforward information of immediate relevance, getting across the point they feel is most important. Can express the main point they want to make comprehensibly.
B2: Can pass on detailed information reliably. Can communicate the essential points even in more demanding situations, though their language lacks expressive power and idiomaticity.
C1: Can qualify opinions and statements precisely in relation to degrees of, for example, certainty/uncertainty, belief/doubt, likelihood, etc. Can make effective use of linguistic modality to signal the strength of a claim, an argument or a position.
C2: Can convey finer shades of meaning precisely by using, with reasonable accuracy, a wide range of qualifying devices (e.g. adverbs expressing degree,

clauses expressing limitations). Can give emphasis, differentiate and eliminate ambiguity.

Coherence and cohesion

A1: Can link words/signs or groups of words/signs with very basic linear connectors (e.g. "and" or "then").

A2: Can use the most frequently occurring connectors to link simple sentences in order to tell a story or describe something as a simple list of points. Can link groups of words/signs with simple connectors (e.g. "and", "but" and "because").

B1: Can introduce a counter-argument in a simple discursive text (e.g. with "however"). Can link a series of shorter, discrete simple elements into a connected, linear sequence of points. Can form longer sentences and link them together using a limited number of cohesive devices, e.g. in a story. Can make simple, logical paragraph breaks in a longer text.

B2: Can use a variety of linking expressions efficiently to mark clearly the relationships between ideas. Can use a limited number of cohesive devices to link their utterances into clear, coherent discourse, though there may be some "jumpiness" in a long contribution. Can produce text that is generally well-organised and coherent, using a range of linking expressions and cohesive devices. Can structure longer texts in clear, logical paragraphs.

C1: Can produce clear, smoothly flowing, well-structured language, showing controlled use of organisational patterns, connectors and cohesive devices. Can produce well-organised, coherent text, using a variety of cohesive devices and organisational patterns.

C2: Can create coherent and cohesive text making full and appropriate use of a variety of organisational patterns and a wide range of cohesive devices.

B Appendix B

I deal with consulting and sales of financial products and services to an international bank, in the mass-market and small-business. I follow the relationship with customers from acquisition to the advise until the realization of contracts, building and maintaining relationships after-sales in the aim of customer satisfaction

I also worked with large and small teams in back-offices, managed many administrative activities related to mortages, personal loans, contability and investments too.

I worked for several years to the acquisition of new customers, to provide them with a complete service, from the account to insurance products, investment products, personal loans, revolving credit, and cross-selling products. In many years of work I have honed my skills in managing non-standard situations, analyzing the problem, finding and implementing practical and easy solutions. non-standard situations, analyzing the problem, finding and implementing practical and easy solutions.

I have faced several situations always work with serenity and enthusiasm, I like to work in a multicultural and dynamic.

I'm careful to meet the goals of the team in which I work, cooperating with colleagues to achieve the goals by providing my skills, always willing to learn, respecting other points of view together finding ways to deal. I work for the same large company for 25 years, now is the time to change and find new job opportunities. Needs to work my husband has been living in Zaandam, I want to find a new job in Holland to rejoin our family.

I like sports such as skiing, riding and swimming. I've also got the rescue licence, I worked as a lifeguard in the summer studying for the patent padi dive master

The holistic score is 3.5 (B1+), and GPT-4 provided these analytic scores:

- general linguistic range: 3
- vocabulary range: 4
- grammatical accuracy: 2
- vocabulary control: 3
- orthographic control: 4
- flexibility: 2
- thematic development: 2
- coherence and cohesion: 1
- propositional precision: 3

C Appendix C

When we include the holistic score, the prompt given to GPT-4 is the following:

```
Consider the following essay:
[ESSAY]
```

> It has been given this score on a scale from 1 to 6.5: [HOLISTIC SCORE].
>
> I want you to assess it only considering the aspect of [ASPECT], for which you have 6 different feedback options, that you will have to accept or reject: [ANALYTIC CEFR DESCRIPTORS]
>
> ONLY ONE option can be accepted and is the option you will have to output by only selecting the option letter in the following format: 'option A/B/C/D/E/F'[10] WITHOUT ANY ADDITIONAL OBSERVATION, COMMENT, NOTE, EXPLANATION, CLARIFICATION, OR JUSTIFICATION OF ANY SORT.
>
> Your answer:

When we do not provide GPT-4 with the holistic score, the prompt is the following:

> Consider the following essay: [ESSAY]
>
> I want you to assess it only considering the aspect of [ASPECT], for which you have 6 different feedback options, that you will have to accept or reject: [ANALYTIC CEFR DESCRIPTORS]
>
> ONLY ONE option can be accepted and is the option you will have to output by only selecting the option letter in the following format: 'option A/B/C/D/E/F'[11] WITHOUT ANY ADDITIONAL OBSERVATION, COMMENT, NOTE, EXPLANATION, CLARIFICATION, OR JUSTIFICATION OF ANY SORT.
>
> Your answer:

D Appendix D

Score alignment

Table 5 shows the holistic score normalisation process for EFCAMDAT.

CEFR	W&I	EFCAMDAT
A1	A1 (1)	1,2
	A1+ (1.5)	3
A2	A2 (2)	4,5
	A2+ (2.5)	6
B1	B1 (3)	7,8
	B1+ (3.5)	9
B2	B2 (4)	10,11
	B2+ (4.5)	12
C1	C1 (5)	13,14
	C1+ (5.5)	15
C2	C2 (6)	16 (score<85)
	C2+ (6.5)	16 (score≥85)

Table 5: Score alignment.

Additional experimental results

Table 6 reports the results of the experiment conducted when no holistic scores are given to GPT-4.

[10] The aspects of vocabulary control, flexibility, and thematic development only have options A-E since no descriptors are available for the A1 level.

[11] See note 10.

	score	%gram.	#dif.wds.	#unq.wds.	%1.d.t.	#unq.n.cks.	#unq.q.m.a.	fl.-kinc.	w2v	av.s.ln.	holistic
Lng.	gen. lin.	0.643	0.622	0.547	0.471	0.526	0.565	0.268	0.275	0.148	0.739
	gramm.	**0.707**	0.408	0.365	0.284	0.324	0.364	0.151	0.170	0.099	0.692
	orth.	**0.730**	0.362	0.290	0.234	0.259	0.309	0.133	0.166	0.068	0.653
	voc. ctrl.	0.697	0.391	0.363	0.305	0.331	0.369	0.153	0.102	0.107	0.654
	voc. rg.	0.529	0.539	0.456	**0.410**	0.450	0.452	**0.247**	0.241	0.131	0.616
Prg.	propos.	0.432	0.510	0.442	0.341	0.430	0.492	**0.246**	**0.304**	0.145	0.539
	coh.	0.602	**0.601**	**0.542**	0.379	**0.533**	**0.571**	0.244	0.299	**0.162**	0.729
	flexib.	0.307	0.361	0.363	0.282	0.348	0.346	0.202	0.149	**0.160**	0.330
	themat.	0.425	**0.612**	**0.587**	**0.496**	**0.576**	**0.583**	0.242	**0.333**	0.150	0.543
	holistic	0.732	0.640	0.665	0.451	0.623	0.637	0.178	0.364	0.141	1.000

Table 6: SRC correlation of the GPT-4 predicted scores and relevant linguistic features (**without giving GPT-4 the holistic score**). The *holistic* entry refers to the ground-truth holistic scores. In bold the two highest correlations columnwise.

Using Program Repair as a Proxy for Language Models' Feedback Ability in Programming Education

Charles Koutcheme and **Nicola Dainese** and **Arto Hellas**
Aalto University, Espoo, Finland
first.last@aalto.fi

Abstract

One of the key challenges in programming education is being able to provide high-quality feedback to learners. Such feedback often includes explanations of the issues in students' programs coupled with suggestions on how to fix these issues. Large language models (LLMs) have recently emerged as valuable tools that can help in this effort. In this article, we explore the relationship between the program repair ability of LLMs and their proficiency in providing natural language explanations of coding mistakes. We outline a benchmarking study that evaluates leading LLMs (including open-source ones) on program repair and explanation tasks. Our experiments study the capabilities of LLMs both on a course level and on a programming concept level, allowing us to assess whether the programming concepts practised in exercises with faulty student programs relate to the performance of the models. Our results highlight that LLMs proficient in repairing student programs tend to provide more complete and accurate natural language explanations of code issues. Overall, these results enhance our understanding of the role and capabilities of LLMs in programming education. Using program repair as a proxy for explanation evaluation opens the door for cost-effective assessment methods.

1 Introduction

Large Language Models (LLMs) and applications leveraging them such as ChatGPT have been embraced by both the general public and academia. The adoption is also visible in the domain of computing and programming education, where researchers have highlighted a variety of learning tasks that LLMs can tackle (Denny et al., 2023; Prather et al., 2023), including their performance in providing help and feedback to students (Hellas et al., 2023).

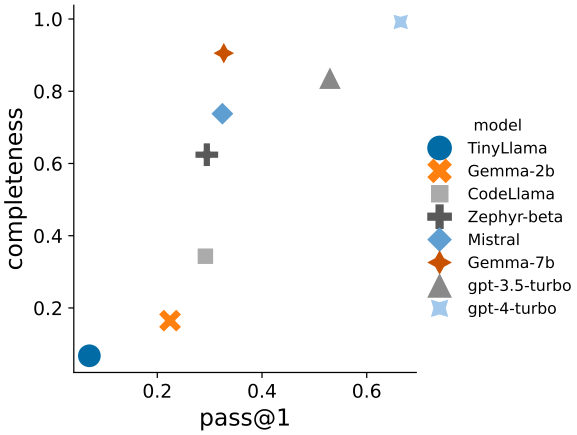


Figure 1: Summary benchmarking results. The quality of LLMs' Natural Language descriptions of issues in students' code (completeness) tends to increase with LLMs' ability to fix the student programs (pass@1).

Feedback is a crucial part of learning (Hattie and Timperley, 2007). While various forms of feedback exist in programming (Keuning et al., 2018), explaining code issues in natural language can be particularly useful. Providing students with natural language explanations of the mistakes in their code allows them to gain a better understanding of gaps in their knowledge.

With the increasing number of LLMs proficient at providing feedback (Koutcheme et al., 2023a) to some degree, selecting the best one before deploying it in classrooms (Liu et al., 2024) can be challenging. Human evaluation can take time, as it requires either manual assessment or annotated datasets. While research in the automated evaluation of LLM generation is on the rise (Zheng et al., 2023), also in educational areas (Fernandez et al., 2024), the developed methods often rely on other language models (e.g., utilizing powerful yet expensive LLMs such as GPT-4), which can induce computational or financial costs. A more cost-effective approach is needed.

Before the advent of LLMs, a stream of work in programming education has focused on educational program repair (Gulwani et al., 2018; Parihar

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 165–181
June 20, 2024 ©2024 Association for Computational Linguistics

et al., 2017; Yi et al., 2017), where the goal is to produce fixes for students' incorrect programs. Although repairs to student programs are not always directly provided to students, they serve as a fundamental step in generating different types of support, including next-step hints for Intelligent Tutoring Systems (McBroom et al., 2021). While direct evaluation of feedback with natural language explanations can be challenging, evaluating whether LLMs can fix programs is much more straightforward.

With this in mind, we hypothesize that the student program repair capability of an LLM may relate to its capability to provide natural language explanations of code issues. If this would hold, program repair capability – which is easier to assess – could serve as a proxy for evaluating feedback quality. Our intuition is supported by prior work that has found relationships between LLMs' abilities in related domains. For instance, LLMs that are proficient in solving specific problems are effective judges of the quality of explanations in those domains (Zheng et al., 2023). Similarly, there is some evidence that instruction-tuned LLMs trained on specific tasks can generalize to unseen parallel or close tasks (Wei et al., 2022a).

In this article, we investigate whether there effectively exists a relationship between the ability of LLMs to repair students' programs and their ability to explain code issues in natural language. If our hypothesis holds, researchers could more easily benchmark LLMs for other educational purposes, allowing educators to streamline the selection of LLMs. Our evaluation focuses on several leading and popular open-source language models, as well as proprietary models.

The main contributions of this article are (1) the benchmarking of several leading language models' abilities for program repair and (2) natural language explanation of code issues, as well as (3) the analysis and identification of the relationship between the two tasks.

2 Related Work

2.1 Program Repair and Feedback

Propagating feedback. Generating natural language explanations of the issues in student programs has been a long-standing challenge, with much work leveraging part of human annotations to bootstrap efforts (Piech et al., 2015; Malik et al., 2021; Koivisto and Hellas, 2022). In that area, early pretrained code language models have also shown useful (Wu et al., 2021) in making human annotations as data efficient as possible. However, coming up with such annotations remains a time-consuming endeavour.

Educational Program Repair. Trying to alleviate the need for manual annotation, feedback on programming assignments has often been generated with the aid of automated program repair tools (Hu et al., 2019a), attempting to repair syntax and/or semantic errors in students' programs. In this area, LLMs have also shown great promise. Much of this line of work has mainly used early versions of the OpenAI Codex model, thus obtaining both syntax fixes (Zhang et al., 2022; Ahmed et al., 2022; Leinonen et al., 2023) and semantic fixes for students' non-working solutions (Zhang et al., 2022). Such fixes can inform Intelligent Tutoring Systems, which could then provide next-step hints to students (Rivers and Koedinger, 2017). However, while automatically constructed next-step hints can tell the students *what* to do next (in templated natural language sentences), they are not always able to explain the reasons *why* the code does not work.

Natural Language Explanations. The rise of newer and more powerful LLMs (e.g., CHATGPT) has opened the possibility of directly generating high-quality code explanations (Sarsa et al., 2022). In addition to such progress, research in improving program repair remains useful. In particular, recent efforts suggest that generated repairs can be included in the prompt to allow language models to provide more accurate natural language explanations of a program's issues (Phung et al., 2023a). In parallel, prior work has also explored using program repair to validate the quality of LLM-generated feedback. In this space, the quality of LLM-generated repairs (i.e., whether the repairs pass all unit tests) would indicate whether the associated LLM-generated feedback would be given to students. The repairs could be generated by the LLM providing the feedback (Shubham Sahai, 2023), or by another, less powerful LLM acting as an artificial student (Phung et al., 2023b).

In contrast to efforts using program repair as a means for validating single generations, our work aims to assess whether the overall ability of a single language model to provide repair across a *larger set of programs* is indicative of the language model's overall ability to generate natural language explanations.

2.2 Evaluating Language Models

Benchmarking code language models. When new language models are released, their performance is often assessed through multiple code generation benchmarks such as HumanEval (Chen et al., 2021), APPS (Hendrycks et al., 2021), MBPP (Austin et al., 2021), or DS-1000 (Lai et al., 2022). In parallel, prior work has also evaluated LLMs' ability to fix buggy programs in benchmarks such as HumanEval+ (Muennighoff et al., 2023), CodeXGlue (Lu et al., 2021), or QuixBugs (Lin et al., 2017). However, while such benchmarks contain multiple tasks that could potentially inform us of LLMs' performance in educational contexts, it is important to note that students' submitted incorrect programs can contain issues/defects that go beyond mere simple bugs (e.g. implementation of the wrong algorithm). Hence, educational benchmarks are needed.

Benchmarking in education. In the educational context, much work has looked into the performance of proprietary models (Codex, and ChatGPT) on private datasets and educational datasets (Finnie-Ansley et al., 2022; Hellas et al., 2023) both for program synthesis (Finnie-Ansley et al., 2022; Savelka et al., 2023b) and feedback (Hellas et al., 2023).

Open-source language models. While there exist few efforts looking at the performance of open-language models for generating repairs (Koutcheme et al., 2023a; Koutcheme, 2023), or answering student programming questions (Hicke et al., 2023), only the work of (Koutcheme et al., 2024) look into the performance of open-source models for generating educational programming feedback. Still, none of these works studies the relationship between program repair abilities and the quality of LLM-generated natural language explanations.

3 Methodology

We (1) evaluate how LLMs perform in generating repairs to incorrect programs, (2) evaluate how LLMs perform in explaining the issues in programs, and (3) study the potential relationship between the ability to generate repairs and the ability to generate natural language explanations. To ensure a comprehensive assessment, our study encompasses zero-shot evaluations (Yogatama et al., 2019; Linzen, 2020) of proprietary and state-of-the-art open-source LLMs having less than 7 billion (7B) parameters. Our experiments leverage a publicly available dataset comprising real-life students' submissions to Python programming problems.

Next, we describe the programming dataset, outline our evaluation methodology, and list the language models included in this evaluation. We release the code used to perform our experiments as an additional contribution [1].

3.1 Dataset

We use a subset of the FalconCode (de Freitas et al., 2023) dataset, a large-scale dataset containing thousands of first-year students' solutions (over three semesters) to hundreds of Python programming assignments. It is the largest and most comprehensive publicly available dataset of student programs at the time of writing this manuscript. Beyond its substantial scale, this dataset distinguishes free-form assignments (i.e., not scoped to function writing), and exercise-level programming with concept annotations, enabling a broader evaluation of LLM feedback.

Dataset processing. Due to the financial and computational costs of running LLM evaluations, for our experiments, we curate a smaller subset of submissions. The dataset contains three semesters worth of submissions (fall 2021, spring 2021, and fall 2022). We start by selecting submissions from the last semester (fall 2022). Each exercise in the dataset can be categorized based on a type (practice, or exam) and a level of difficulty ("skill", "lab", or "project", i.e., easy, medium, hard). We omit exam exercises and focus on practice exercises (as these are the ones students require help with). Additionally, we exclude more complex "project" assignments, requiring extensive code writing across multiple files, and those requiring external files. Following Hu et al. (2019b), we select only the final incorrect submissions for each student for each assignment. Although this selection may not capture the full range of student difficulties, it aligns with the idea that a student's last attempt often reflects their final understanding. Finally, we remove submissions with identical abstract syntax tree structures after variable normalization (Koutcheme et al., 2023a,c). The final dataset contains 370 programs from 44 assignments.

[1] https://github.com/KoutchemeCharles/bea2024

3.2 Repairing Student Programs

Given a student's incorrect program in our test set, the first task is for an LLM to produce a repair to that incorrect program that passes all unit tests. Because of the wide range of issues found in students' programs, in contrast to classical program repair benchmarks (Lin et al., 2017; Muennighoff et al., 2023), in most educational scenarios, we do not assume the existence of a single unique ground truth repair to an incorrect program. However, while such unique ground truth does not exist, repairs that align with the original incorrect programs are often preferred. The general assumption is that closely aligned programs can generate (Phung et al., 2023a) or are associated with feedback (Koutcheme et al., 2023a) (e.g. natural language explanations or hints) that are more understandable to students, as this feedback would require a lower cognitive load to understand the issues in the program and the modifications that need to be operated to reach a solution (Shubham Sahai, 2023). Moreover, we aim to investigate whether the language model's ability to produce repairs that closely resemble the original incorrect programs correlates with its proficiency in generating complete and accurate natural language explanations of the issues in the programs. The constraints on functional correctness and closeness are reflected in our evaluation procedure, which we adapt from the work of Koutcheme et al. (2023a).

Evaluation procedure. To evaluate functional correctness, for each incorrect program in our test set, we generate a single repair using greedy decoding (Rozière et al., 2023). To measure the ability of the language model to generate close repairs, we compute the ROUGE-L (Lin, 2004) score between the incorrect program and the candidate repair extracted from the single greedy generation. While other distance measures exist and have been used to measure closeness between programs (e.g., BLEU (Papineni et al., 2002) and CodeBERT score (Zhou et al., 2023b)), the ROUGE-L score has been shown to correlate well with human judgement of high-quality repairs (Koutcheme et al., 2023b) while remaining fast to compute, as it does not rely on a language model.

We report the average repair success rate as the pass rate ('pass@1' (Chen et al., 2021)) and the average ROUGE-L score, abbreviated as 'rouge', over the programs in our test set.

3.3 Explaining Issues in Students Programs

The second task is for our language models to explain all the issues in a given student's incorrect program. For each incorrect program, we prompt our language model to explain the issues using the prompt shown in Figure 5 (Appendix A.1), a variant of the prompt used in (Hellas et al., 2023). Following prior work, we generate a single output using greedy decoding (Hellas et al., 2023; Savelka et al., 2023a; Leinonen et al., 2023).

Evaluation criteria. For each natural language explanation, we focus on two particular quantitative aspects of quality: (1) ensuring that the feedback is complete, i.e., it identifies and mentions all issues in the code, and (2) ensuring that it avoids hallucinations, i.e., it does not mention non-existent issues (Phung et al., 2023b; Hicke et al., 2023; Hellas et al., 2023). We highlight that our explanation task is a specific form of feedback that differs from hints. In the explanation task, the answer is meant to be given to students, while for hints (Roest et al., 2024), the feedback *helps* the students find the answer themselves. While prior work in hint generation has investigated other qualitative aspects, such as the "right level of detail"((Phung et al., 2023a; Scarlatos et al., 2024)), we believe these are less likely to be correlated with an LLM repair ability.

Automated Evaluation. Given the scale of our dataset and the multitude of language models to assess, conducting human evaluation would be impractical. Therefore, we rely on automated evaluation using language models (Zheng et al., 2023). Powerful language models like ChatGPT have exhibited near-human performance across various tasks, sparking interest in their application for evaluating other LLMs (Zhou et al., 2023a; Cui et al., 2023; Tunstall et al., 2023), including in educational contexts (McNichols et al., 2024; Hicke et al., 2023). Notably, GPT-4 has demonstrated good performance in evaluating programming feedback quality (Koutcheme et al., 2024). In our work, we ask GPT-4 to grade the quality of the natural language explanations for each incorrect program. We ask the model to provide a binary label of whether each criterion (completeness, and avoiding highlighting non-existent issues) holds for the feedback generated by each language model. Figure 6 (appendix A.1) shows our prompt. For each criterion, we report the average over the test set.

3.4 Models

We focus our evaluation on instruction-tuned and chat models. While pretrained language models can also be useful for multiple tasks, as prior studies using Codex (Phung et al., 2023a) have shown, instruction-tuned models alleviate the need for complex queries and allow easier interactions which benefit educators and researchers.

Closed-source models. We evaluate GPT-3.5 (`gpt-3.5-turbo`) and GPT-4-turbo (`gpt-4-1106-preview`) on our two tasks. Due to the financial costs of running GPT-4, we use the Turbo version for feedback generation, but we keep the standard GPT-4 for evaluating the quality of the natural language generations.

Open-source models. While prior work in programming feedback using LLMs has focused mainly on ChatGPT models (i.e., GPT-3.5 and GPT-4), we aim to cover the wider range of available options and include a selected number of instruction-tuned open-source/permissive models. We report the performance of the following family of models:

- TinyLLama (Zhang et al., 2024), a 1.1B parameter model following the Llama (Touvron et al., 2023) architecture.
- CodeLLAMA (Rozière et al., 2023), series of Llama (Touvron et al., 2023) models specialized for code. We report the performance of the 7B parameters model.
- Mistral 7B (Jiang et al., 2023), a 7B parameters language model released by the MistralAI team.
- Zephyr (Tunstall et al., 2023) are 7B parameters language models fine-tuned by HuggingFace using Direct Preference Optimization (Rafailov et al., 2023) on top of Mistral 7B model. We evaluate the performance of Zephyr 7B β.
- Gemma (Google, 2024), open source model released by Google DeepMind. We evaluate the performance of the 2B and 7B parameters models.

We chose these families of models because they are fully open-source and well-documented, they perform competitively on various code benchmarks (for models of their size), and they are widely adopted in the community. Additionally, within these families, we choose language models having 7 billion parameters or less, as such models can generally fit within one large GPU (without quantization). This choice is reflected by the potential need for educators to run models on custom hardware, who are unlikely to have the computational and financial resources to access more than a single GPU.

Table 1: Summary of the performance of the models in program repair and code issue explanation. For the metrics pass@1, rouge, and completeness, a higher score indicates better performance. Conversely, for the hallucination rate metric, a lower score is preferable. Legend: compl. (completeness), hall. rate (hallucination rate).

model	repair		explanation	
	pass@1	rouge	compl.	hall. rate (↓)
TinyLlama	0.070	0.062	0.068	0.335
Gemma-2b	0.224	0.175	0.165	0.400
CodeLlama	0.292	0.251	0.343	0.841
Zephyr-beta	0.295	0.236	0.624	0.716
Mistral	0.324	0.241	0.738	0.397
Gemma-7b	0.327	0.298	0.905	**0.005**
gpt-3.5-turbo	0.530	0.470	0.838	0.368
gpt-4-turbo	**0.665**	**0.536**	**0.992**	0.024

Technical details. We query ChatGPT models using OpenAI's Python API. We run the selected open-source language models using the HuggingFace Transformers library (Wolf et al., 2020), each model is run on a single NVIDIA A100 using our institution research cluster. We run all models using their recommended precision. The details of each model (the names) can be found in Table 3 (appendix A.2).

4 Results

First, we describe our general results, then, we outline an ablation analysis detailing the performance of the selected models over a set of programming concepts.

4.1 Main Results

Table 1 summarizes the performance of the LLMs in program repair and in explaining issues in code. We can make the following observations:

LLMs proficient in program repair generate repairs closer to the original incorrect program.

Figure 2 highlights the scaling relationship between the pass rate and the rouge score. We see that as language models become more and more proficient in generating repairs, these repairs become closer to students' original programs and thus more useful. One could expect that LLMs which produce more fixes could generate generic solutions (which are far away from the student code) (Koutcheme et al., 2023c) – however, this is not the case.

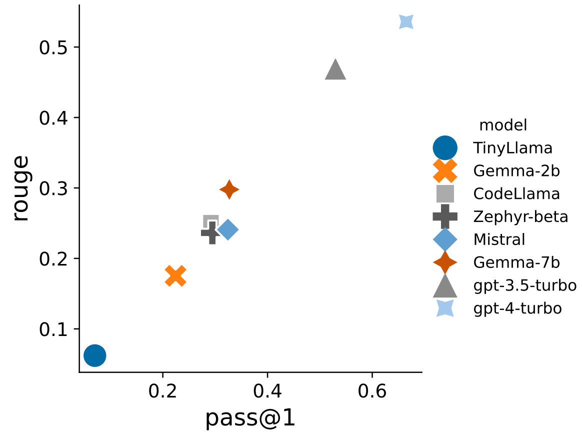


Figure 2: Relationship between pass rate and rouge score.

Hallucination conditionally decreases as a function of completeness. Figure 3 highlights the relationship between the ability of a model to identify all issues in a program (completeness), and the model's tendency to hallucinate (hallucination rate). If we omit language models with less than 2B parameters (i.e., TinyLlama and Gemma-2B), we observe that the hallucination rate decreases as completeness increases. This relationship seems to hold only for large enough language models. Our interpretation is supported by prior work that has shown that many emerging behaviours in language models appear when sufficiently large sizes are reached (Wei et al., 2022b) (e.g. their ability to solve new tasks via chain-of-thought prompting (Wei et al., 2023)).

The ability to explain moderately scales with the ability to repair. Figure 1 highlights the relationship between repair performance and explanation performance (in terms of completeness). Generally, a language model that is better at program repair tends to also produce more complete descriptions. In the set of our LLMs, only Gemma-7B and GPT-3.5 disrupt this relationship: although Gemma-7B has a lower pass rate than GPT-3.5 (only slightly better than Mistral), it produces very complete explanations (and with fewer hallucinations). Interestingly, the performance gap between models' ability to repair does not reflect the gap between their ability to explain in natural language. For instance, the difference between CodeLLama and Zephyr-7B in pass@1 (0.003) is almost 10 × smaller than the performance gap between the models' abilities to generate complete explanations (0.281).

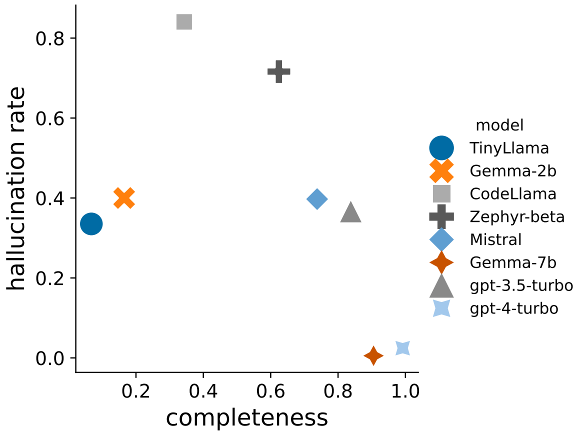


Figure 3: Relationship between completeness and hallucination rate.

Reparing student programs is harder than explaining issues in natural language. When looking at the maximum value that the pass@1 metric assumes (0.665), we see that it is smaller than the one of the completeness (0.992). We believe repairing programs is more challenging than providing explanations, as the latter requires understanding the issues while the former requires both comprehension and expertise on how to implement the fixes.

On base models and fine-tuning. We hypothesize that pass@1 and completeness are reflective of the capabilities of the underlying base model, while the hallucination rate seems to depend more on the fine-tuning procedure. Our intuition is justified by the following observations: (1) Mistral and Zephyr share the same base model (but only different fine-tuning) and have comparable pass@1 and completeness, but very different hallucination rates. OpenAI and Google invest significant efforts into curating datasets for fine-tuning to avoid hallucinations. On the other hand, small language models (TinyLLama and Gemma-2b) are probably too inaccurate (i.e., not powerful enough) to even hallucinate.

4.2 Concept Level Performance Analysis

The FalconCode dataset contains information about 20 programming concepts or "skills" (e.g., function definition, assignment, conditionals). The authors of the dataset manually annotated each exercise with information on whether each of these skills is practised (or needs to be mastered) in each exercise. We refer the reader to the original paper for details about the concepts (de Freitas et al., 2023).

In the same way that some students exhibit varying struggles with understanding and practising specific programming concepts (Liu et al., 2023), we suspect that language models might face a similar challenge. By examining the performance of language models on a per-concept basis, we aim to provide insights into their strengths and weaknesses in addressing specific programming challenges, thus informing educators and developers on their suitable application scenarios.

We thus conduct an ablation study looking at the per-concept performance of our language models for repair and natural language explanation generation.

Methodology. For each of the 20 concepts, we obtain the list of exercises practising the concept and subsequently retrieve the incorrect programs in our test set submitted to these exercises. For each concept, we then report and compare the performance of the language models for program repair and natural language generation (using the same evaluation metrics) based on the retrieved subset of incorrect programs.

It is important to note that because all exercises practice multiple concepts, knowing which *individual* concept is responsible for the language model failing to fix (or explain) the issues in a program is impossible. As such, the following results will give us an overview of the *likelihood* that an LLM would struggle to support students if an exercise *involves* such a concept. Table 4 (Appendix A.3) shows the number of exercises and programs that practice each specific concept. We limit our analysis to concepts practised in more than 3 exercises.

Results. Due to space limitations, we focus our analysis on the concepts with which language models struggle the most. Table 2 shows these concepts for all performance metrics, which are derived from Table 6 in Appendix B.2 showing the detailed scores of all models. We can make the following observations:

Table 2: Programming concepts performance summary. We show the programming concept for which each language model struggles the most. Legend: IS (input string), IC (input casting), C (conditionals), FC (function call), FD (function definition), L (list), LU (loop until), L2D (list 2D), hall. rate (hallucination rate).

	pass@1	rouge	completeness	hall. rate
TinyLlama	IC	IC	LU	IS
Gemma-2b	LU	IS	LU	L2D
CodeLlama	IC	IC	L2D	L
Zephyr-beta	IS	IS	FD	C
Mistral	IS	IS	FD	L
Gemma-7b	IS	IS	FC	LU
gpt-3.5-turbo	IC	IC	LU	FC
gpt-4-turbo	IS	IS	LU	L2D

When looking into the worst-practised concepts for repairing student programs, almost all of them are related to input manipulation (input string, or input casting), similar to what has been observed in LLMs capability to provide suggestions to programming help requests (Hellas et al., 2023). Moreover, LLMs that perform poorly at fixing a given concept are also likely to perform poorly at generating close solutions for these concepts.

When looking at the worst concepts for natural language explanations, these concern a wider range (looping, data structure, functions, basic operations). For completeness, there is not much variation in the performance in explaining issues for different concepts, but rather the overall performance is correlated with the pass@1 of the corresponding model. For hallucination rate, each model has its own “base performance”, which doesn’t correlate with pass@1 and it’s roughly constant across concepts, with the exceptions of Zephyr and gpt-3.5-turbo, which respectively over- and underperform on function-related concepts concerning other concepts. There is no clear association between the concepts where LLMs are accurate and those where they hallucinate. Both small language models (less than 7B parameters) and proprietary models struggle most to be accurate with the ‘looping until’ concept, while language models of 7B parameters struggle more with function-related assignments.

It is important to note that “struggling” here is relative to the model’s performance with other concepts. GPT-4 “struggling” more on completeness with looping is still accurate 90% of the time.

5 Discussion

Repair as a proxy for feedback. Our results suggest that language models' relative ability to fix students' programs (which is easy to evaluate) tells us how these language models will compare in finding all issues in students' code while avoiding hallucination (for big enough language models). Based on our discovery, one can devise more efficient LLM selection pipelines. For instance, a simple strategy consists of filtering out language models for which repair performance does not reach a certain threshold, a threshold set based on a few evaluations of LLM natural language generation performance. As an illustrative scenario, only evaluating the Mistral model on our dataset allows us to reasonably assume that language models performing worse than 0.32 in pass rate (pass@1) are unlikely to generate complete explanations for more than 73.8 % of programs. Using this pass rate value can thus act as a selection lower bound. As LLMs are becoming more widely adopted in education (Prather et al., 2023; Denny et al., 2024), and as the number of available models is increasing, these insights can help in the adoption process as institutions evaluating LLMs for their context can potentially reduce the number of LLMs to consider or limit the number of tasks conducted during the evaluation.

Open-source language models strike back. Another important finding emerging from our results is that while high-performance program repair must rely on proprietary models, recent 7B parameters models such as Gemma-7B can generate high-quality feedback competitive with SOTA models (Koutcheme et al., 2024). This has positive implications for educators interested primarily in giving students feedback rather than repairing solutions, as such feedback can also be generated via privacy-preserving open-source models.

However, it's important to acknowledge that running such models requires custom computational resources. In the literature, 7B parameter models are sometimes termed "small" due to their relative size compared to many large language models (e.g. Falcon-180B model (Almazrouei et al., 2023)). Yet, a 7B parameter is not small in terms of computational resources as it requires a large GPU to fit entirely into memory (without quantization). There is currently a trend in developing small language models (less than 3B parameters) such as TinyLlama and Gemma which can run on more modest hardware (e.g., consumer laptop GPU, or accelerated hardware). However, the performance of such LLMs, as our results suggest is still lagging behind their 7B parameters counterparts.

Identifying specific knowledge gaps. Unfortunately, our results do not yet allow us to identify which programming concepts LLMs will struggle to explain in natural language from their program repair performance. While individual repair performance depends on the concept being practised, a language model's performance in explaining issues does not (i.e., the performance is constant across all concepts). We hypothesize that the per-concept performance gap is only revealed for the harder task of fixing students' programs. Uncovering LLM knowledge gaps with automated measures might require us to rely on harder automatically evaluable tasks (e.g. QLCs (Lehtinen et al., 2024)).

Interplay of programming feedback types. Our primary research objective is to deepen our understanding of LLMs' feedback capabilities in educational contexts. Specifically, we seek to explore the relationship between different forms of feedback and program repair. While we treated feedback (identifying and explaining issues in programs) and program repair as distinct tasks in this study, we acknowledge their inherent interdependence. Previous research suggests that high-quality repairs can induce high-quality feedback when provided in context (Phung et al., 2023b,a). However, generating high-quality repairs is inherently challenging, as our results suggest, requiring the language model to comprehend what is wrong in a program and how to address the issues. In contrast, we believe explanations of issues in students' programs could serve as reasoning steps (Wei et al., 2023), enhancing the subsequent generation of repairs (Chen et al., 2023). These refined repairs, in turn, could facilitate the generation of high-quality next-step hints (Roest et al., 2024). Research investigating the interplay between different types of feedback is thus pivotal in unlocking the full potential of language models to support programming education. By studying the performance of generating repairs without conditioning on feedback, nor generating feedback based on repairs, our work establishes a foundational understanding that will allow the research community to assess the extent to which various prompting techniques enhance feedback performance.

6 Conclusions

In this article, we have uncovered an intriguing relationship between LLM performance in program repair and the capability to explain issues in code. Our evaluations encompassed both open-source and proprietary models, examining their generic performance as well as concept-specific proficiency.

While selecting and deploying a specific language model may not be challenging, identifying the most suitable one for a particular purpose can be complex, particularly when considering financial, hardware, or other limitations. At a time when there are calls to rethink how programming is taught (Denny et al., 2024), the insights gleaned from our work can provide valuable guidance for educators in choosing LLMs that align with their instructional contexts.

Future work. Our future work will involve two specific directions. First, we'll continue our investigation of the relationships between various types of programming feedback and program repair. all these efforts remain an attempt to streamline the selection process of language models based on automated evaluation measures.

Besides studying LLM performance, our second objective is to leverage our computational resources to improve these LLMs' ability to provide feedback. In particular, small language models' poor explaining performance suggests that these models will benefit from alignment procedures designed specifically to improve feedback abilities (Scarlatos et al., 2024).

Limitations

Our work is not free of limitations. We evaluated the LLMs on a subset of solutions from a single dataset (from one institution with one programming language). Moreover, our evaluation of natural language explanations relied on GPT-4, which, although a state-of-the-art language model, is not a perfect evaluator. Human evaluation is necessary to strengthen our results. Furthermore, refinement would benefit the evaluation prompt (e.g., allowing GPT-4 to reason (Wei et al., 2023) before providing its final answers). Additionally, the results of our evaluation also depend on the specific prompts used to interact with each language model. Similarly, our benchmarking experiment was not exhaustive – although we included many popular state-of-the-art open-source and proprietary models, many more exist. Including more models would be necessary to strengthen the claim of the relationship between repair and natural language explanations. Beyond this, the concept analysis is only indicative, as many assignments feature multiple concepts. Finally, we only considered single-turn zero-shot repair, which does not take advantage of LLMs' ability to reason with few-shot examples (Brown et al., 2020), or LLMs' ability to correct their own mistakes (Chen et al., 2023; Xia and Zhang, 2023).

Ethics Statement

The work in the present article has been conducted following national and institutional ethics guidelines. We recognize the increasing importance of ethical considerations in artificial intelligence research, particularly concerning data usage and potential societal impacts.

The dataset employed in this research is openly available to researchers. Our overarching goal is to contribute to the development and evaluation of open-source language models for providing feedback in programming education. By focusing on open-source models, we aim to promote transparency, accessibility, and accountability in AI research and development, thereby addressing concerns regarding the privacy implications of using proprietary language models.

We further acknowledge the broader ethical implications of our work, including issues related to fairness and accessibility of LLM feedback, how LLMs might favour certain styles of interaction, and how LLMs might contribute to inequalities in the quality of provided education worldwide.

References

Toufique Ahmed, Noah Rose Ledesma, and Premkumar Devanbu. 2022. Synfix: Automatically fixing syntax errors using compiler diagnostics.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The falcon series of open language models.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and

Charles Sutton. 2021. Program synthesis with large language models.

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.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating language models trained on code.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. 2023. Teaching large language models to self-debug.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback.

Adrian de Freitas, Joel Coffman, Michelle de Freitas, Justin Wilson, and Troy Weingart. 2023. Falconcode: A multiyear dataset of python code samples from an introductory computer science course. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, SIGCSE 2023, page 938–944, New York, NY, USA. Association for Computing Machinery.

Paul Denny, James Prather, Brett A Becker, James Finnie-Ansley, Arto Hellas, Juho Leinonen, Andrew Luxton-Reilly, Brent N Reeves, Eddie Antonio Santos, and Sami Sarsa. 2023. Computing education in the era of generative ai. *arXiv preprint arXiv:2306.02608*.

Paul Denny, James Prather, Brett A. Becker, James Finnie-Ansley, Arto Hellas, Juho Leinonen, Andrew Luxton-Reilly, Brent N. Reeves, Eddie Antonio Santos, and Sami Sarsa. 2024. Computing education in the era of generative ai. *Commun. ACM*, 67(2):56–67.

Nigel Fernandez, Alexander Scarlatos, and Andrew Lan. 2024. Syllabusqa: A course logistics question answering dataset.

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 *Proceedings of the 24th Australasian Computing Education Conference*, ACE '22, page 10–19, New York, NY, USA. Association for Computing Machinery.

Google. 2024. Gemma: Open models based on gemini research and technology. Technical report, Google DeepMind. Https://storage.googleapis.com/deepmind-media/gemma/gemma-report.pdf.

Sumit Gulwani, Ivan Radiček, and Florian Zuleger. 2018. Automated Clustering and Program Repair for Introductory Programming Assignments. ArXiv:1603.03165 [cs].

John Hattie and Helen Timperley. 2007. The power of feedback. *Review of educational research*, 77(1):81–112.

Arto Hellas, Juho Leinonen, Sami Sarsa, Charles Koutcheme, Lilja Kujanpää, and Juha Sorva. 2023. Exploring the responses of large language models to beginner programmers' help requests. In *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 1*, ICER '23, page 93–105, New York, NY, USA. Association for Computing Machinery.

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.

Yann Hicke, Anmol Agarwal, Qianou Ma, and Paul Denny. 2023. Ai-ta: Towards an intelligent question-answer teaching assistant using open-source llms.

Yang Hu, Umair Z. Ahmed, Sergey Mechtaev, Ben Leong, and Abhik Roychoudhury. 2019a. Refactoring based program repair applied to programming assignments. In *2019 34th IEEE/ACM Int. Conf. on Automated Software Engineering (ASE)*, pages 388–398. IEEE/ACM.

Yang Hu, Umair Z. Ahmed, Sergey Mechtaev, Ben Leong, and Abhik Roychoudhury. 2019b. Refactoring based program repair applied to programming assignments. In *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 388–398. IEEE/ACM.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Hieke Keuning, Johan Jeuring, and Bastiaan Heeren. 2018. A systematic literature review of automated feedback generation for programming exercises. *ACM Transactions on Computing Education (TOCE)*, 19(1):1–43.

Teemu Koivisto and Arto Hellas. 2022. Evaluating codeclusters for effectively providing feedback on code submissions. In *2022 IEEE Frontiers in Education Conference (FIE)*, pages 1–9. IEEE.

Charles Koutcheme. 2023. Training Language Models for Programming Feedback Using Automated Repair Tools. In *Artificial Intelligence in Education*, pages 830–835, Cham. Springer Nature Switzerland.

Charles Koutcheme, Nicola Dainese, Sami Sarsa, Arto Hellas, Juho Leinonen, and Paul Denny. 2024. Open source language models can provide feedback: Evaluating llms' ability to help students using gpt-4-as-a-judge. In *Proceedings of the 2024 Innovation and Technology in Computer Science Education, Volume 1*, ITICSE '24.

Charles Koutcheme, Nicola Dainese, Sami Sarsa, Juho Leinonen, Arto Hellas, and Paul Denny. 2023a. Benchmarking educational program repair. In *NeurIPS'23 Workshop on Generative AI for Education (GAIED)*. NeurIPS.

Charles Koutcheme, Sami Sarsa, Juho Leinonen, Lassi Haaranen, and Arto Hellas. 2023b. Evaluating distance measures for program repair. In *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 1*, ICER '23, page 495–507, New York, NY, USA. Association for Computing Machinery.

Charles Koutcheme, Sami Sarsa, Juho Leinonen, Arto Hellas, and Paul Denny. 2023c. Automated Program Repair Using Generative Models for Code Infilling. In *Artificial Intelligence in Education*, pages 798–803, Cham. Springer Nature Switzerland.

Yuhang Lai, Chengxi Li, Yiming Wang, Tianyi Zhang, Ruiqi Zhong, Luke Zettlemoyer, Scott Wen tau Yih, Daniel Fried, Sida Wang, and Tao Yu. 2022. Ds-1000: A natural and reliable benchmark for data science code generation.

Teemu Lehtinen, Charles Koutcheme, and Arto Hellas. 2024. Let's ask ai about their programs: Exploring chatgpt's answers to program comprehension questions. In *Proceedings of the 46th International Conference on Software Engineering: Software Engineering Education and Training*, ICSE-SEET '24.

Juho Leinonen, Arto Hellas, Sami Sarsa, Brent Reeves, Paul Denny, James Prather, and Brett A. Becker. 2023. Using language models to enhance programming error messages. In *Proceedings of the 2023 ACM SIGCSE Technical Symposium on Computer Science Education*.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro, Oleh Shliazhko, Nicolas Gontier, Nicholas Meade, Armel Zebaze, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Benjamin Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy, Jason Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nour Fahmy, Urvashi Bhattacharyya, Wenhao Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Kunakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailey Schoelkopf, Jan Ebert, Tri Dao, Mayank Mishra, Alex Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. 2023. Starcoder: may the source be with you!

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.

Derrick Lin, James Koppel, Angela Chen, and Armando Solar-Lezama. 2017. Quixbugs: A multi-lingual program repair benchmark set based on the quixey challenge. In *Proceedings Companion of the 2017 ACM SIGPLAN International Conference on Systems, Programming, Languages, and Applications: Software for Humanity*, SPLASH Companion 2017, page 55–56, New York, NY, USA. Association for Computing Machinery.

Tal Linzen. 2020. How can we accelerate progress towards human-like linguistic generalization? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, Online. Association for Computational Linguistics.

Qi Liu, Shuanghong Shen, Zhenya Huang, Enhong Chen, and Yonghe Zheng. 2023. A survey of knowledge tracing.

Rongxin Liu, Carter Zenke, Charlie Liu, Andrew Holmes, Patrick Thornton, and David J. Malan. 2024. Teaching cs50 with ai: Leveraging generative artificial intelligence in computer science education. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, SIGCSE 2024, page 750–756, New York, NY, USA. Association for Computing Machinery.

Shuai Lu, Daya Guo, Shuo Ren, Junjie Huang, Alexey Svyatkovskiy, Ambrosio Blanco, Colin Clement, Dawn Drain, Daxin Jiang, Duyu Tang, Ge Li, Lidong Zhou, Linjun Shou, Long Zhou, Michele Tufano, Ming Gong, Ming Zhou, Nan Duan, Neel Sundaresan, Shao Kun Deng, Shengyu Fu, and Shujie Liu. 2021. Codexglue: A machine learning benchmark dataset for code understanding and generation.

Ali Malik, Mike Wu, Vrinda Vasavada, Jinpeng Song, Madison Coots, John Mitchell, Noah Goodman, and Chris Piech. 2021. Generative Grading: Near Human-level Accuracy for Automated Feedback on Richly Structured Problems. In *Proceedings of the 14th Educational Data Mining conference*.

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.

Hunter McNichols, Wanyong Feng, Jaewook Lee, Alexander Scarlatos, Digory Smith, Simon Woodhead, and Andrew Lan. 2024. Automated distractor and feedback generation for math multiple-choice questions via in-context learning.

Niklas Muennighoff, Qian Liu, Armel Zebaze, Qinkai Zheng, Binyuan Hui, Terry Yue Zhuo, Swayam Singh, Xiangru Tang, Leandro von Werra, and Shayne Longpre. 2023. Octopack: Instruction tuning code large language models.

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.

Sagar Parihar, Ziyaan Dadachanji, Praveen Kumar Singh, Rajdeep Das, Amey Karkare, and Arnab Bhattacharya. 2017. Automatic grading and feedback using program repair for introductory programming courses. In *Proceedings of the 2017 ACM conference on innovation and technology in computer science education*, pages 92–97.

Tung Phung, José Cambronero, Sumit Gulwani, Tobias Kohn, Rupak Majumdar, Adish Singla, and Gustavo Soares. 2023a. Generating high-precision feedback for programming syntax errors using language models.

Tung Phung, Victor-Alexandru Pădurean, Anjali Singh, Christopher Brooks, José Cambronero, Sumit Gulwani, Adish Singla, and Gustavo Soares. 2023b. Automating human tutor-style programming feedback: Leveraging gpt-4 tutor model for hint generation and gpt-3.5 student model for hint validation.

Chris Piech, Jonathan Huang, Andy Nguyen, Mike Phulsuksombati, Mehran Sahami, and Leonidas Guibas. 2015. Learning program embeddings to propagate feedback on student code.

James Prather, Paul Denny, Juho Leinonen, Brett A Becker, Ibrahim Albluwi, Michelle Craig, Hieke Keuning, Natalie Kiesler, Tobias Kohn, Andrew Luxton-Reilly, et al. 2023. The robots are here: Navigating the generative ai revolution in computing education. In *Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education*, pages 108–159.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model.

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(1):37–64.

Lianne Roest, Hieke Keuning, and Johan Jeuring. 2024. Next-step hint generation for introductory programming using large language models. In *Proceedings of the 26th Australasian Computing Education Conference*, ACE '24, page 144–153, New York, NY, USA. Association for Computing Machinery.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2023. Code llama: Open foundation models for code.

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*, pages 27–43.

Jaromir Savelka, Arav Agarwal, Marshall An, Chris Bogart, and Majd Sakr. 2023a. Thrilled by your progress! large language models (gpt-4) no longer struggle to pass assessments in higher education programming courses. In *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 1*, ICER '23, page 78–92, New York, NY, USA. Association for Computing Machinery.

Jaromir Savelka, Arav Agarwal, Christopher Bogart, Yifan Song, and Majd Sakr. 2023b. Can generative pretrained transformers (gpt) pass assessments in higher education programming courses? *arXiv preprint*.

Alexander Scarlatos, Digory Smith, Simon Woodhead, and Andrew Lan. 2024. Improving the validity of automatically generated feedback via reinforcement learning.

Ben Leong Shubham Sahai, Umair Z. Ahmed. 2023. Improving the coverage of gpt for automated feedback on high school programming assignments. In *NeurIPS'23 Workshop on Generative AI for Education (GAIED)*. NeurIPS.

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.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of lm alignment.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022a. Finetuned language models are zero-shot learners.

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. 2022b. Emergent abilities of large 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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Mike Wu, Noah D. Goodman, Chris Piech, and Chelsea Finn. 2021. Prototransformer: A meta-learning approach to providing student feedback. *CoRR*, abs/2107.14035.

Chunqiu Steven Xia and Lingming Zhang. 2023. Conversational automated program repair.

Jooyong Yi, Umair Z Ahmed, Amey Karkare, Shin Hwei Tan, and Abhik Roychoudhury. 2017. A feasibility study of using automated program repair for introductory programming assignments. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, pages 740–751.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, and Phil Blunsom. 2019. Learning and evaluating general linguistic intelligence.

Jialu Zhang, José Cambronero, Sumit Gulwani, Vu Le, Ruzica Piskac, Gustavo Soares, and Gust Verbruggen. 2022. Repairing bugs in python assignments using language models.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023a. Lima: Less is more for alignment.

Shuyan Zhou, Uri Alon, Sumit Agarwal, and Graham Neubig. 2023b. CodeBERTScore: Evaluating code generation with pretrained models of code. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13921–13937, Singapore. Association for Computational Linguistics.

A Experiment details

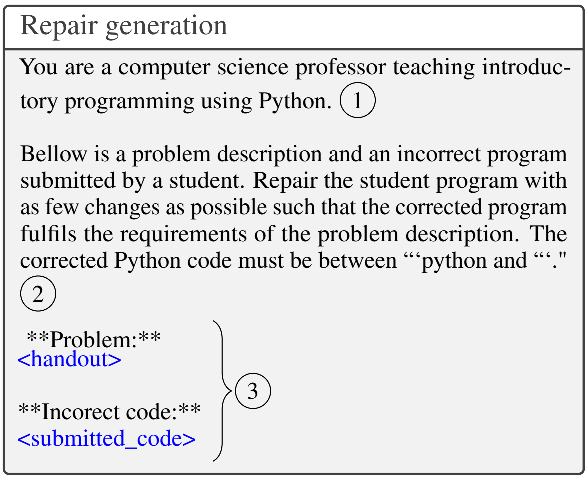
A.1 Prompts used

Figure 4 (resp. Figure 5) shows our prompts to obtain repairs (resp. feedback) from the language models. Figure 6 shows the prompt used to grade the feedback generated by the language models using GPT-4 as our automatic evaluator (we adapt the prompt from (Koutcheme et al., 2024)). The reported value for "completeness" corresponds to the proportion of "yes" responses across our test dataset to the first criterion, while the reported value for the hallucination rate corresponds to the proportion of "no" responses to the second criterion. We note that regarding the issues present in the students' incorrect program, we assumed them to be identified by GPT-4 during evaluation (without a separate prompt). We acknowledge the limitations of this prompting strategy (i.e., no space for reasoning) which we'll refine in future work.

A.2 Official model names

Table 3 translates each model name into their Huggface id [2].

[2] https://huggingface.co/models



Repair generation

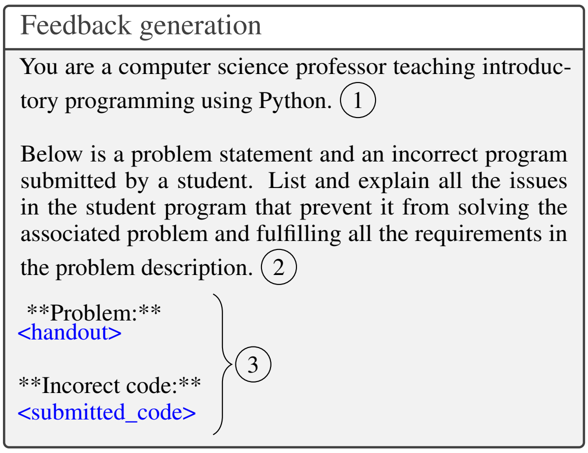
You are a computer science professor teaching introductory programming using Python. ①

Bellow is a problem description and an incorrect program submitted by a student. Repair the student program with as few changes as possible such that the corrected program fulfils the requirements of the problem description. The corrected Python code must be between “‘python and “‘." ②

Problem:
<handout>

Incorect code:
<submitted_code>

Figure 4: Our template for prompting the LLMs to provide feedback. (1) A system prompt specifying the behaviour of the model. (2) A description of the grading task. (3) Information necessary to grade the feedback.



Feedback generation

You are a computer science professor teaching introductory programming using Python. ①

Below is a problem statement and an incorrect program submitted by a student. List and explain all the issues in the student program that prevent it from solving the associated problem and fulfilling all the requirements in the problem description. ②

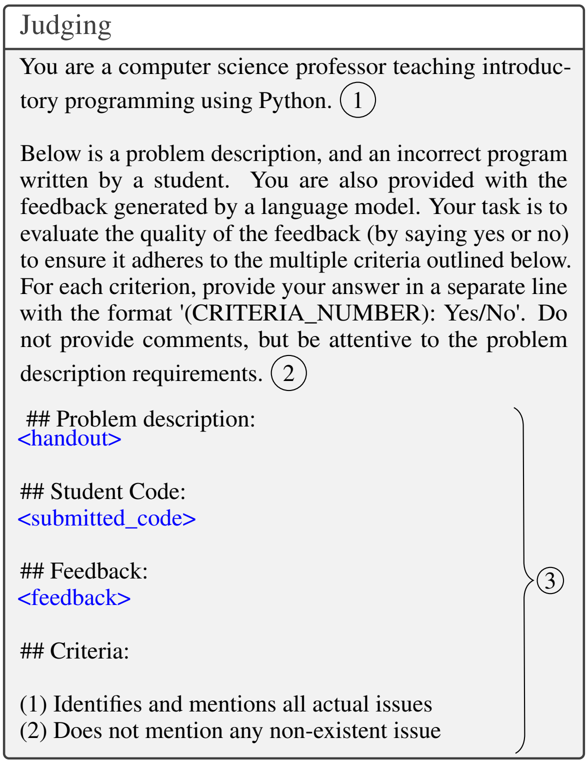
Problem:
<handout>

Incorect code:
<submitted_code>

Figure 5: Our template for prompting the LLMs to provide feedback. (1) A system prompt specifying the behaviour of the model. (2) A description of the grading task. (3) Information necessary to grade the feedback.

Table 3: Official model names for HuggingFace models.

name	HuggingFace/OpenAI id
TinyLlama	TinyLlama/TinyLlama-1.1B-Chat-v1.0
CodeLlama	codellama/CodeLlama-7b-hf
Llama	meta-llama/Llama-2-7b-chat-hf
Mistral	mistralai/Mistral-7B-v0.1
Zephyr	HuggingFaceH4/zephyr-7b-beta
Gemma	google/gemma-7b-it



Judging

You are a computer science professor teaching introductory programming using Python. ①

Below is a problem description, and an incorrect program written by a student. You are also provided with the feedback generated by a language model. Your task is to evaluate the quality of the feedback (by saying yes or no) to ensure it adheres to the multiple criteria outlined below. For each criterion, provide your answer in a separate line with the format '(CRITERIA_NUMBER): Yes/No'. Do not provide comments, but be attentive to the problem description requirements. ②

## Problem description:
<handout>

## Student Code:
<submitted_code>

## Feedback:
<feedback>

## Criteria:

(1) Identifies and mentions all actual issues
(2) Does not mention any non-existent issue

Figure 6: Judging prompt template. We provide (1) a system prompt specifying GPT-4's behaviour, (2) a description of the grading task, and (3) contextual information.

A.3 Concept analysis

Table 4 shows the number of exercises which practice each concept. Additionally, figure 7 shows an upset plot of the number of incorrect programs for which each combination of programming concepts is practised.

B Results details

B.1 Additional performance scores

Some work in program synthesis has evaluated the ability of language models to generate programs using another method to estimate pass@1. This method, originally proposed in the work of Chen et al. (Chen et al., 2021), is based on generating multiple samples, and is particularly adapted to non-instruction tuned models. We report the results of the program repair performance evaluation based on this multi-sample strategy.

Multi-sample performance evaluation. For each incorrect program, we generate $n = 20$ samples using top_p nucleus sampling and a temperature of 0.2 (Chen et al., 2021; Li et al., 2023). We evaluate functional correctness using the pass@1

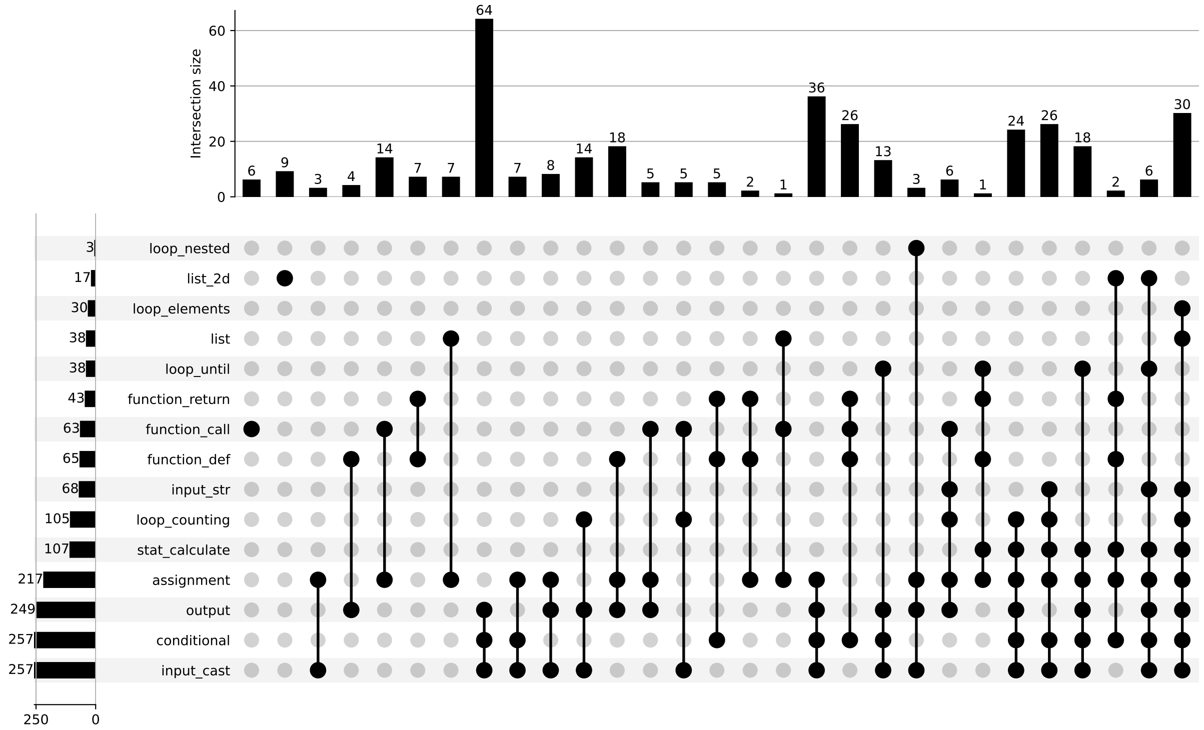


Figure 7: Programming concepts upset plot.

Table 4: Number of exercises and incorrect programs practised for each concept.

concept	# exercises	# programs
input string	4	18
input casting	27	257
output	28	249
assignment	26	217
conditional	22	257
function calling	8	63
function definition	9	65
function return	6	43
loop counting	9	105
loop until	5	38
loop elements	1	30
loop nested	1	3
stat calculation	10	38
list	3	38
list 2D	3	17

estimator, which tells us the probability that a language model will fix an incorrect program in a single attempt (Muennighoff et al., 2023).

To evaluate the ability of a language model to generate a solution close to the student program, we average the ROUGE-L score between each of the $k(k \leq n)$ candidate repairs that pass all unit tests and the incorrect program.

Results. Table 5 shows the performance results with the adapted pass@1 and rouge scores for a subset of the models (those with more than 7B parameters).

Table 5: We show the pass@1, rouge, completeness, and hallucination rate (hall. rate).

model	pass@1	rouge	completeness	hall. rate
Gemma-7b	0.267	0.353	0.905	0.005
Zephyr-beta	0.276	0.336	0.624	0.716
Mistral	0.304	0.365	0.738	0.397
gpt-3.5-turbo	0.529	0.561	0.838	0.368
gpt-4-turbo	0.634	0.559	0.992	0.024

In general, we notice an absolute drop in performance from the greedy decoding. Beyond this absolute difference, the main change is that the ranking of the model changed. Gemma-7B is now the least performant of the 7B parameters models.

The performance of the 7B parameters model are dependent on these.

B.2 Programming concepts performance

Table 6 shows the detailed per concept performance results for all models.

Table 6: Per concept performance results. Legend: IS (input string), IC (input casting), O (output), A (assignment), C (conditionals), FC (function call), FD (function definition), FR (function read), LC (loop counting), LU (loop until), SC (stat calculate), L (list), L2D (list 2D).

(a) Pass@1

	IS	IC	O	A	C	FC	FD	FR	LC	LU	SC	L	L2D
TinyLlama	0.04	0.03	0.03	0.07	0.05	0.21	0.09	0.14	0.03	0.03	0.04	0.03	0.06
Gemma-2b	0.06	0.13	0.13	0.12	0.19	0.44	0.60	0.77	0.13	0.05	0.10	0.11	0.47
CodeLlama	0.22	0.18	0.24	0.24	0.26	0.54	0.52	0.56	0.19	0.29	0.21	0.24	0.59
Zephyr-beta	0.10	0.17	0.17	0.24	0.25	0.60	0.58	0.86	0.23	0.26	0.26	0.26	0.41
Mistral	0.13	0.23	0.22	0.27	0.28	0.56	0.49	0.67	0.19	0.24	0.24	0.34	0.65
Gemma-7b	0.16	0.22	0.25	0.29	0.25	0.52	0.52	0.53	0.21	0.47	0.21	0.26	0.47
gpt-3.5-turbo	0.44	0.41	0.50	0.52	0.46	0.84	0.86	0.91	0.49	0.50	0.55	0.68	0.76
gpt-4-turbo	0.21	0.58	0.63	0.64	0.63	0.86	0.92	1.00	0.39	0.50	0.48	0.42	0.76
average	0.17	0.24	0.27	0.30	0.30	0.57	0.57	0.68	0.23	0.29	0.26	0.29	0.52

(b) Completeness

	IS	IC	O	A	C	FC	FD	FR	LC	LU	SC	L	L2D
TinyLlama	0.04	0.07	0.07	0.04	0.08	0.03	0.08	0.07	0.04	0.00	0.05	0.08	0.18
Gemma-2b	0.15	0.15	0.14	0.17	0.15	0.21	0.20	0.26	0.16	0.05	0.17	0.18	0.06
CodeLlama	0.31	0.33	0.32	0.37	0.35	0.43	0.35	0.35	0.33	0.45	0.42	0.26	0.24
Zephyr-beta	0.54	0.64	0.65	0.59	0.63	0.60	0.51	0.51	0.55	0.71	0.60	0.76	0.59
Mistral	0.81	0.74	0.71	0.77	0.75	0.79	0.69	0.79	0.73	0.76	0.79	0.79	0.82
Gemma-7b	0.94	0.94	0.92	0.91	0.95	0.76	0.86	0.91	0.90	1.00	0.94	0.95	1.00
gpt-3.5-turbo	0.93	0.82	0.82	0.87	0.84	0.84	0.86	0.81	0.83	0.76	0.89	0.97	0.94
gpt-4-turbo	1.00	0.99	0.99	1.00	0.99	0.98	0.98	0.98	0.99	0.97	1.00	1.00	1.00
average	0.59	0.58	0.58	0.59	0.59	0.58	0.57	0.58	0.57	0.59	0.61	0.62	0.60

(c) ROUGE

	IS	IC	O	A	C	FC	FD	FR	LC	LU	SC	L	L2D
TinyLlama	0.04	0.03	0.03	0.07	0.05	0.17	0.08	0.13	0.03	0.03	0.04	0.03	0.06
Gemma-2b	0.05	0.11	0.11	0.11	0.15	0.32	0.45	0.57	0.12	0.05	0.10	0.09	0.31
CodeLlama	0.20	0.15	0.20	0.22	0.21	0.46	0.45	0.47	0.17	0.25	0.18	0.21	0.51
Zephyr-beta	0.07	0.13	0.13	0.20	0.19	0.50	0.48	0.71	0.18	0.21	0.21	0.22	0.32
Mistral	0.08	0.15	0.16	0.20	0.20	0.44	0.38	0.52	0.14	0.15	0.16	0.26	0.48
Gemma-7b	0.16	0.20	0.22	0.28	0.23	0.49	0.47	0.49	0.20	0.43	0.21	0.25	0.44
gpt-3.5-turbo	0.41	0.37	0.44	0.47	0.41	0.74	0.76	0.80	0.45	0.45	0.51	0.63	0.72
gpt-4-turbo	0.17	0.46	0.51	0.52	0.50	0.70	0.72	0.77	0.31	0.40	0.38	0.35	0.61
average	0.15	0.20	0.22	0.26	0.24	0.48	0.47	0.56	0.20	0.25	0.22	0.26	0.43

(d) hallucination rate

	IS	IC	O	A	C	FC	FD	FR	LC	LU	SC	L	L2D
TinyLlama	0.47	0.30	0.25	0.35	0.31	0.41	0.42	0.40	0.43	0.21	0.40	0.39	0.18
Gemma-2b	0.12	0.37	0.42	0.37	0.35	0.32	0.48	0.40	0.24	0.55	0.28	0.13	0.65
CodeLlama	0.87	0.86	0.87	0.82	0.85	0.75	0.82	0.81	0.82	0.89	0.80	0.97	0.88
Zephyr-beta	0.88	0.79	0.78	0.80	0.78	0.46	0.60	0.49	0.86	0.61	0.89	0.87	0.65
Mistral	0.43	0.42	0.42	0.39	0.41	0.32	0.32	0.35	0.41	0.32	0.37	0.45	0.41
Gemma-7b	0.00	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.01	0.03	0.02	0.00	0.00
gpt-3.5-turbo	0.28	0.31	0.35	0.36	0.35	0.54	0.49	0.51	0.34	0.21	0.29	0.24	0.24
gpt-4-turbo	0.00	0.02	0.02	0.02	0.02	0.00	0.03	0.05	0.00	0.05	0.01	0.05	0.06
average	0.38	0.38	0.39	0.39	0.38	0.35	0.40	0.38	0.39	0.36	0.38	0.39	0.38

Automated Evaluation of Teacher Encouragement of Student-to-Student Interactions in a Simulated Classroom Discussion

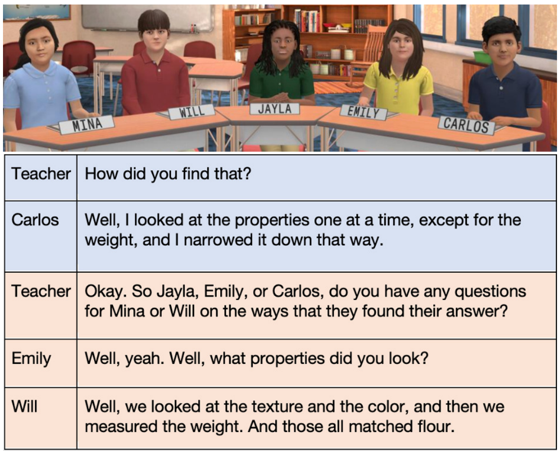
Michael John Ilagan
McGill University
michael.ilagan@mail.mcgill.ca

Beata Beigman Klebanov
ETS Research Institute
bbeigmanklebanov@ets.org

Jamie N. Mikeska
ETS Research Institute
jmikeska@ets.org

Abstract

Leading students to engage in argumentation-focused discussions is a challenge for elementary school teachers, as doing so requires facilitating group discussions with student-to-student interaction. The Mystery Powder (MP) Task was designed to be used in online simulated classrooms to develop teachers' skill in facilitating small group science discussions. In order to provide timely and scaleable feedback to teachers facilitating a discussion in the simulated classroom, we employ a hybrid modeling approach that successfully combines fine-tuned large language models with features capturing important elements of the discourse dynamic to evaluate MP discussion transcripts. To our knowledge, this is the first application of a hybrid model to automate evaluation of teacher discourse.



Teacher	How did you find that?
Carlos	Well, I looked at the properties one at a time, except for the weight, and I narrowed it down that way.
Teacher	Okay. So Jayla, Emily, or Carlos, do you have any questions for Mina or Will on the ways that they found their answer?
Emily	Well, yeah. Well, what properties did you look?
Will	Well, we looked at the texture and the color, and then we measured the weight. And those all matched flour.

Figure 1: The Mursion Upper Elementary Classroom Environment, with an excerpt from a Mystery Powder discussion transcript. Two blocks of utterances (explained in section 4.2) are shown in blue and orange, respectively. Image provided by Mursion, Inc.

1 Introduction

Scientific argumentation is an essential skill, and in elementary school classrooms, group science discussions are a natural modality for providing students with opportunities to engage in scientific argumentation (Sampson and Blanchard, 2012; Shemwell and Furtak, 2010). Accordingly, it is essential that teachers are well equipped to facilitate such discussions. But facilitating them is not straightforward. Many teachers are used to a lecture style of interaction where they deliver the facts and the students respond only to the teacher (Cazden, 1988; Lemke, 1990; Lloyd et al., 2016). In contrast, in an ideal group science discussion, students directly interact with their peers (rather than just the teacher) and engage with each other's ideas, rather than only their own and the teacher's (Fishman et al., 2017; Tenenbaum et al., 2020).

Digitally simulated classroom experiences have become increasingly used to prepare teachers for the work of teaching (Dalinger et al., 2020; Dieker et al., 2014). In a simulated classroom, the teacher-in-training, also called a *pre-service teacher* (henceforth, **PST**), enacts a classroom scenario, interacting in real time with student avatars puppeteered by a trained human actor equipped with voice modulating software. In contrast to practicum experiences, simulated classrooms afford development of targeted skills in an environment that is both standardized and low-stakes (Dalinger et al., 2020; Bondie et al., 2021; Cohen et al., 2020; Ersozlu et al., 2021). Automating as much as possible of the simulation would help make the learning experience more affordable and thus accessible to a wider range of teachers; it would also allow teachers to engage in multiple rounds of practice to hone their teaching skills. Of the two bottlenecks—the puppeteer enacting the student avatars and the human expert evaluating the performance—we here address the second, leaving the first to future work.

The present paper is a case study of developing automated evaluation, with supervised learning, of a PST's performance in a simulated classroom. We focus on the Mystery Powder (henceforth, **MP**)

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 182–198
June 20, 2024 ©2024 Association for Computational Linguistics

task (Mikeska et al., 2021), a particular lesson that the PST is to teach in a simulated classroom (Figure 1) designed to develop PSTs' competency in facilitating small group argumentation-focused science discussions at the elementary level. Successful facilitation of a discussion is complex; in this work, we address one of its dimensions, namely, the extent to which the teacher encourages student-to-student interactions where students engage directly with each other's ideas (Mikeska et al., 2021; GO Discuss Project, 2021).

In line with the manual evaluation process that produced the training data (Mikeska et al., 2019), our approach to automated evaluation had models on two levels: classifiers identifying PST utterances as positive examples of the desired teaching practices; and regressors scoring the transcript as a whole on the same practices (Nazaretsky et al., 2023). Furthermore, we kept in mind two considerations: classifier training must deal with the fact that rater labels were non-exhaustive (only some utterances are labeled); and regressors must aggregate utterance-level information in an intuitive way.

In terms of what features were used, we built three types of models: (a) models based on the analysis of the content of what the PST said, implemented using fine-tuned large language models (henceforth, **LLM**s); (b) models based on the structure of the interaction, who speaks when and in relation to whose utterance; and (c) combined models using both content and structure. To our knowledge, this is the first demonstration, in the context of automated analysis of teacher discourse, of a successful combination of fined-tuned LLMs and shallow features into a hybrid model that outperforms both components in isolation across the board, for multiple levels of analysis (utterance-level and transcript-level) and multiple indicators of performance.

2 Related Work

2.1 Elements of high-quality teaching practices

Recent research has addressed automated detection of high-quality teaching practices in human-annotated corpora of real classroom transcripts. Demszky and colleagues (Demszky et al., 2021; Demszky and Hill, 2023; Alic et al., 2022) detected features associated with dialogic instruction, such as teachers' conversational uptake (Demszky et al., 2021) and open-ended questions (Alic et al., 2022), which they found to benefit classroom outcomes such as student satisfaction and participation. Similar discourse features were investigated in Jensen et al. (2020), as part of an effort to bring easy-to-use and high-quality audio recording setups to ordinary classrooms. Suresh and colleagues (Suresh et al., 2019, 2022b) performed a six-way classification of teacher utterances into discursive strategies, called "talk moves" (e.g. "Keeping everyone together"), that promote equitable student participation. Tran et al. (2023) classified student and teacher contributions into 'talk moves' such as 'teacher links student contributions' and 'students support claims with evidence'. Nazaretsky et al. (2023) studied ways to evaluate to what extent participants provided meaningful contributions that moved the discussion forward. Most of the prior work, with few exceptions such as Nazaretsky et al. (2023), considered transcripts of live interactions; simulated environments with student avatars aim to extend the practice earlier into the teacher preparation process, before the teacher meets a real classroom (Dalinger et al., 2020). Our work is in the much less explored context of a simulated classroom.

A common theme in research on automated models for high-quality teaching practice is the intended application to providing automated feedback to teachers. Feedback may come in the form of a dashboard summarizing the teacher's performance. The dashboard may report the (relative) frequency of the target discourse features (Demszky et al., 2023; Jensen et al., 2020). The dashboard may also cite "positive examples" among the teacher's own utterances to reinforce productive teaching practices (Demszky et al., 2023; Jensen et al., 2020; Nazaretsky et al., 2023). The efficacy of such automated feedback for benefiting classroom outcomes (e.g. proportion of assignments completed by the student) has been demonstrated in a setting with 1:10 teacher-student ratio (Demszky et al., 2023) as well for 1:1 mentoring (Demszky and Liu, 2023).

2.2 Modeling Approaches

In terms of modeling approaches, prior work explored pre-trained deep neural embeddings to represent the content of an utterance and either use them directly as features for detecting teachers' discourse moves of interest (Suresh et al., 2019) or to derive features such as similarity scores between neighboring teacher and student utterances when modeling uptake (Demszky et al., 2021). Demszky et al. (2021) reported that simpler lexical features quantifying token overlap between student and teacher words were also competitive. Jensen et al. (2020) used a combination of linguistic features such as parts of speech and markers of comparisons or definitions along with features capturing other characteristics of the teacher-student interaction, including utterance length and its normalized position in the session, rate of speech and pauses, in a supervised machine learning setting.

Fine-tuning an LLM-based classifier for the target data and task was also explored. Jensen et al. (2021) found the performance of a BERT-based classifier to be superior to that of feature-based baselines on data of self-recorded classroom interactions from English Language Arts teachers. Nazaretsky et al. (2023) fine-tuned DistilBERT (Sanh et al., 2020) on simulated classroom data in the science domain. Suresh et al. (2021) explored BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) to classify student and teacher utterances into 'talk moves' in the domain of mathematics. Tran et al. (2023) used a sequence model BiLSTEM-CRF (Huang et al., 2015) with BERT embeddings to classify utterances into a somewhat different set of 'talk moves' in the domain of English Language Arts and showed that the sequence model that takes into account neighboring utterances outperformed the BERT-based models that did not utilize sequence information, for detecting some of the talk moves. Suresh et al. (2022b) explored incorporating information from the discourse outside of the teacher's and neighboring student utterance, showing that taking a much larger discourse context into account helps improve performance; the best-performing models extended the context to seven prior and seven subsequent utterances. Kumaran et al. (2023) explored fine-tuning of DialoGPT (Zhang et al., 2020), a dialog LLM built on GPT-2 (Radford et al., 2019) on the student subset of the 'talk moves' data (Suresh et al., 2022a), utilizing a context of nine prior utterances. These approaches tend to include elements of the larger discourse context through incorporation of larger and larger chunks of prior and/or subsequent content into the LLM-based framework.

In the present study, we explore an approach that models discourse dynamics more directly through a set of features that would be used in tandem with the fine-tuned LLM to provide the overall model with information about relevant aspects of the structure of the discourse. Such hybrid models can also provide some insights into the task, by separating the contribution of the fine-tuned LLM based content models from that of the discourse-dynamic-based model; different aspects may be more or less important for modeling different components of the complex performance task set to the teachers.

3 The Mystery Powder Task

3.1 The performance

In the MP task, the PST interacts with five upper elementary student avatars in the simulated classroom (Figure 1). Each avatar is standardized, in terms of their personality (e.g. Will is soft-spoken) and preconceptions related to the MP task (explained below). The human actor, who puppeteers all five avatars, is well-versed in them and is instructed to ensure that they are responsive to the PST's instructions throughout the discussion.

The scenario is as follows. Prior to the discussion, the class was shown samples of six powders: flour, cornstarch, baking soda, baking powder, sugar, and salt. The class investigated several properties of each sample including texture, color, weight, reaction with vinegar, and outcome when mixed with water. Subsequently, the class was presented a "mystery powder" sample—in fact baking soda, unbeknowst to the students—and the students investigated its properties as well. In small groups, as pre-work to the discussion, the students reflected in writing on their findings and generated evidence-based claims about the mystery powder's identity and the properties that were useful to identify the mystery powder. See Appendix for a reference table for the powders (Figure 6) and an excerpt from one of the group's pre-work (Figure 7).

The PST has up to 20 minutes to facilitate a discussion to help the five students arrive at a consensus regarding (1) the identity of the mystery powder, and (2) which properties are important for this identification. As preparation, the PST has access to the students' written reflections and is provided information about the accuracy of their initial ideas. For instance, the PST must ensure that the discussion rectifies the misconception (held by Mina, Will, Jayla, and Emily) that weight is important for identifying the MP. See Figure 1 for an excerpt from a discussion's transcript.

Dimension 3: Encouraging Student-to-Student Interactions

Indicator title	Level 1 Beginning practice	Level 2 Developing practice	Level 3 Well-prepared practice
3a. Peer interaction	The teacher assumes the responsibility for the discussion by rarely promoting peer interaction AND frequently mediates all student contributions.	The teacher occasionally promotes peer interaction, AND the majority of student contributions are mediated through the teacher.	The teacher frequently promotes peer interaction, AND the mediation of student contributions is shared between the teacher and the students.
3b. Engagement with others' ideas	The teacher rarely encourages students to engage with one another's ideas, conceptions, or viewpoints.	The teacher occasionally encourages students to engage with one another's ideas, conceptions, or viewpoints.	The teacher frequently encourages students to engage with one another's ideas, conceptions, or viewpoints.

Table 1: Rubrics for Indicators 3A and 3B (Mikeska et al., 2021).

3.2 Rubric and manual evaluation

The MP rubric is made up of several dimension scores, each of which is supported by several more specific indicator scores (Mikeska et al., 2021). The present study focuses on Dimension 3 ("Encouraging student-to-student interactions") and two of its indicators, Indicator 3A ("Peer interaction") and Indicator 3B ("Engagement with other's ideas").[1] See Table 1 for Indicator score definitions.

After evaluation, the PST expects to see a feedback report that tells their strengths, areas for growth, and recommended next steps in each Dimension. This report must give not only an overall (i.e. transcript-level) evaluation but also supporting evidence (i.e. utterance-level) to reinforce the PST's desirable practices.

Accordingly, manual evaluation occurs on two levels. First, the human rater cites, for each Indicator, one or more utterances that exemplify the target behavior (positive examples) or its opposite (negative examples). Note that the rater is asked only to provide some examples, not exhaustively label every utterance in the transcript. Second, the human rater scores the transcript, continuous on a scale of 1 to 3 (e.g. a score of 1.40 is possible) on each Indicator and then an integer from 1 to 3 for each Dimension. To calibrate judgments, raters undergo extensive training, which includes completing self-guided webinars and evaluating sample discussions.

3.3 Automated evaluation approach

Automated evaluation aims to follow the same two-level process, via classifiers (for utterances) and regressors (for transcripts). Conceptually, regressor features are aggregates of utterance-level information, which include utterance class labels. However, ground-truth labels are not available for new transcripts, so aggregating them is infeasible. Instead, in our approach, after training on the labeled utterances, a classifier predicts positive probabilities for all utterances, labeled and unlabeled. It is then these predicted probabilities that are aggregated into transcript-level features (described in section 4.2). Thus, classifier training and evaluation uses ground-truth labels, for the subset of utterances they are available; but regressor training and evaluation uses only imputed probabilities.

4 Data, models, and features

4.1 Data

The MP dataset was collected in prior work (Mikeska et al., 2019).[2] A total of 79 PSTs facilitated discussions: 76 engaged in the simulation twice; 3 engaged once. Of the 155 transcripts, 56 were coded by two raters. Reliability was measured with intra-class correlation coefficients (ICCs) and was sufficient (Cicchetti, 1994) for all three constructs: 0.816 for Indicator 3A; 0.679 for Indicator 3B; 0.635 for Dimension 3. For transcripts scored by two raters, the final scores were the average between the raters—thus non-integer scores are also possible for Dimension 3. The MP dataset has a total of 14,558 utterances. For PSTs (6,713 utterances), the interquartile range for utterance length was 8 to 30 tokens; for students (7,845 utterances), it was 4 to 20 tokens. Distributions of transcript-

[1] Dimension 3 has a third indicator, "Ideas come from students", not within the scope of the present study.

[2] The Mystery Powder discussion and scoring data used in this study was collected and generated as part of previous grants funded by the National Science Foundation (grant #1621344, #2032179, and #2037983). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

level scores are in Figure 2.

Allocation of transcripts into train and test partitions was done by PST, so that PSTs in the training data would not be again seen in the test set (Nazaretsky et al., 2023). 121 transcripts (from 62 PSTs) were allocated to the training set and 34 transcripts to the test set. For utterance-level analyses, each utterance was allocated to the same partition (train or test) as its parent transcript.

	Indicator 3A		Indicator 3B	
	Train	Test	Train	Test
Class 0	1411	668	558	179
Class 1	267	86	426	144
(unlabeled)	3496	785	4190	1216

Table 2: Breakdown of PST utterances by class label, construct, and train/test.

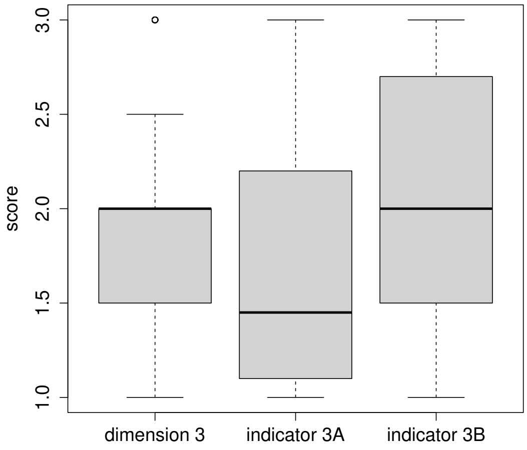


Figure 2: Distributions of transcript scores (training set).

Raters' citations of positive examples were in free-form text, which was manually coded by the first author. The test set was coded after model selection on the training set. Rater comments were not always timestamps or direct quotes, so some judgment was exercised. The following rules were applied:

1. A PST utterance is labeled "1" (positive) if at least one rater cited it as a positive example.

2. If a PST utterance is not labeled "1", then it is labeled "0" (nonpositive) if at least one rater indicated that the transcript had no positive examples in it.

3. If a PST utterance is not labeled "1", then it is labeled "0" (nonpositive) if at least one rater indicated that it was a negative example. Since there were only a few negative examples, they were not assigned their own class.

4. If a PST utterance cannot be labeled either "0" or "1" due to the above rules, it is left unlabeled—excluded from training and evaluation of the utterance-level classifiers.

Note that for training and evaluation of classifiers, only the manually-labeled PST utterances are used. But for training and evaluation of regressors, all PST utterances are used, as predicted probabilities are used instead of ground-truth labels.

Since PST performance is the focus of the study, student utterances were used only to generate features pertaining to the adjacent PST utterances, following a process explained in section 4.2. See Table 2 for breakdown of the PST utterance labels in the dataset. Only a small proportion of the utterances are positive examples (in the training set, 5% for Indicator 3A and 8% for Indicator 3B).

4.2 Models and features

As we inspected rater justifications and rubric definitions, we decided to hand-craft a number of features as well as leverage neural language models shown to be useful in prior work on teacher discourse analysis (see Section 2). In all, we considered 15 models, summarized in Table 3. Models vary along three factors — level of analysis, target construct, and type of features, as follows:

- A model is either (**C**) an utterance level classifier, or (**R**) a transcript-level regressor.

- A model is concerned with (**A**) Indicator 3A, (**B**) Indicator 3B, or (**D**) Dimension 3.

- A model is (**N**) content-based (via fine-tuning an LLM), (**S**) structure-based (via handcrafted features, some of which involve using LLMs out of the box), or (**X**) a combination of both.

Note that only Indicators have utterance-level analysis, so there are no classifiers for Dimension 3. Also note that models "compete" only in the same cell (e.g. CAN vs. CAS vs. CAX).

Content-only classifiers (CAN and CBN) were constructed by adding a linear classifier head on

	(A) Indicator 3A	(B) Indicator 3B	(D) Dimension 3
(C) Utterance-level classifier	**CAN**: content only **CAS**: structure only **CAX**: combined	**CBN**: content only **CBS**: structure only **CBX**: combined	(none)
(R) Transcript-level regressor	**RAN**: content only **RAS**: structure only **RAX**: combined	**RBN**: content only **RBS**: structure only **RBX**: combined	**RDN**: content only **RDS**: structure only **RDX**: combined

Table 3: All models. See the beginning of section 4.2 for an explanation of the rows and columns.

top of DistilBERT (Sanh et al., 2020) (66M parameters) using the HuggingFace toolkit (Wolf et al., 2020). DistilBERT is a lightweight model that has been used in educational settings (Nazaretsky et al., 2023; Datta et al., 2023; Butt et al., 2022; Pearce et al., 2023). Embedding and transformer layers were frozen. Training was done with learning rate 0.001, batch size 32, and a linear scheduler with no warmup. The number of epochs (between 1 and 10) was a hyperparameter. As inputs to DistilBERT, each utterance was prepended by the speaker (e.g. "Carlos"), and the context for each PST utterance was the student utterance immediately following the teacher's in the transcript. The intuition is that how students respond is potentially informative for whether the PST utterance is positive or not.

Unlike fine-tuning an LLM, which leverages utterance content, classifiers with handcrafted features mostly use turn-taking dynamics, that is, the structure of the interaction. Utterances (student and PST) are organized in blocks. Each PST utterance begins a block, which spans the subsequent student utterances until the next PST utterance. Figure 1 shows two color-coded blocks of utterances. By computing features per block, features associated with a PST utterance incorporate the turn-taking structure in the subsequent student utterances.

For the structure-only classifier for Indicator 3A (CAS), the following four features were computed per PST utterance based on its block:

- NUM_STUDTURNS: Number of student utterances in the block.
- NUM_TEACHTOKS: Number of tokens in the PST utterance itself.
- NUM_STUDTOKS: Number of tokens in the students' utterances in the block.
- NUM_KW1: "1" if the tokens "turn" and "talk" both appear in the PST utterance; "1" if the token "crosstalk" appears in the PST utterance; and "0" otherwise. ("Turn and talk" is the name of a commonly-used instructional technique where students are put in pairs to discuss an issue (Hindman et al., 2022). In the case of the MP discussion, when this occurs, the avatars produce mumbling sounds often denoted in the transcript as "crosstalk".)

For the structure-only classifier for Indicator 3B (CBS), the handcrafted features capture student-to-student uptake. Each student utterance u_1 is paired with the previous student utterance u_0 in the transcript. For every such pairing, the following five features are computed:

- PROP_IN_LEFT: Proportion of tokens in u_0 also found in u_1, range: [0,1].
- PROP_IN_RIGHT: Proportion of tokens in u_1 also found in u_0, range: [0,1].
- JACCARD: Jaccard coefficient between the two sets of tokens, range: [0,1].
- BLEU: BLEU (Papineni et al., 2002) score for reference u_0 and hypothesis u_1, range: [0,1].
- SENTBERT: Cosine similarity between the sentence-BERT (Reimers and Gurevych, 2019) embeddings of u_0 and u_1, range: [-1,1].

Classifiers		Regressors	
(LR)	Logistic regression	(LR)	Linear regression
(DT)	Decision tree	(BR)	Bayesian ridge regression
(MP)	Multilayer perceptron	(DT)	Decision tree
(RF)	Random forest	(MP)	Multilayer perceptron
		(RF)	Random forest

Table 4: Classifiers and regressors to choose from.

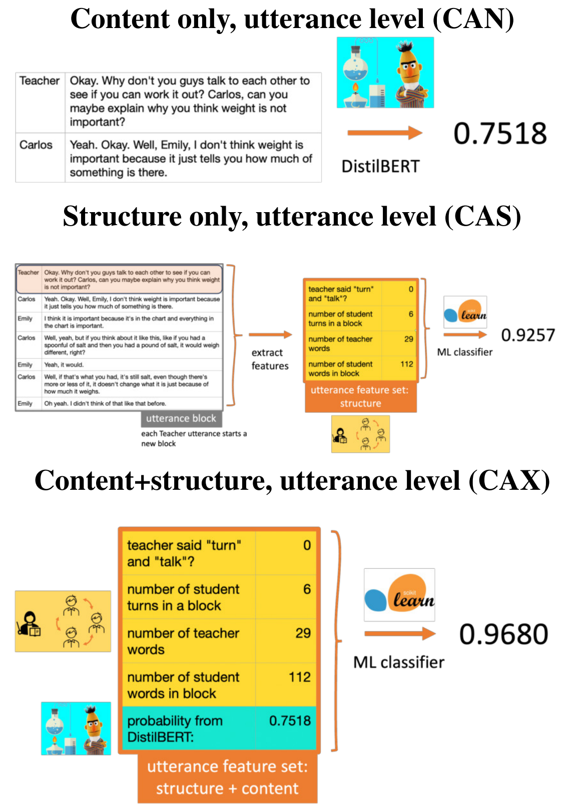

Content only, utterance level (CAN)
Teacher
Okay. Why don't you guys talk to each other to see if you can work it out? Carlos, can you maybe explain why you think weight is not important?
Carlos
Yeah. Okay. Well, Emily, I don't think weight is important because it just tells you how much of something is there.
DistilBERT
0.7518
Structure only, utterance level (CAS)
Teacher
Okay. Why don't you guys talk to each other to see if you can work it out? Carlos, can you maybe explain why you think weight is not important?
Carlos
Yeah. Okay. Well, Emily, I don't think weight is important because it just tells you how much of something is there.
Emily
I think it is important because it's in the chart and everything in the chart is important.
Carlos
Well, yeah, but if you think about it like this, like if you had a spoonful of salt and then you had a pound of salt, it would weigh different, right?
Emily
Yeah, it would.
Carlos
Well, if that's what you had, it's still salt, even though there's more or less of it, it doesn't change what it is just because of how much it weighs.
Emily
Oh yeah. I didn't think of that like that before.
utterance block
each Teacher utterance starts a new block
extract features
teacher said "turn" and "talk"? 0
number of student turns in a block 6
number of teacher words 29
number of student words in block 112
utterance feature set: structure
ML classifier
0.9257
Content+structure, utterance level (CAX)
teacher said "turn" and "talk"? 0
number of student turns in a block 6
number of teacher words 29
number of student words in block 112
probability from DistilBERT: 0.7518
utterance feature set: structure + content
ML classifier
0.9680

Figure 3: Illustration of utterance-level modeling in Indicator 3A, for a single PST utterance. Refer to Table 3 for model acronyms. Indicator 3B models (CBN, CBS, CBX) proceed analogously. Structure features are highlighted in yellow; content feature is highlighted in turquoise.

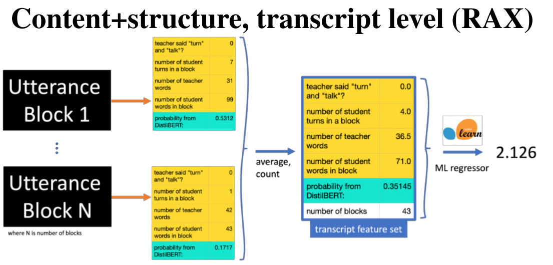

Content+structure, transcript level (RAX)
Utterance Block 1
teacher said "turn" and "talk"? 0
number of student turns in a block 7
number of teacher words 31
number of student words in block 99
probability from DistilBERT: 0.5312
Utterance Block N
where N is number of blocks
teacher said "turn" and "talk"? 0
number of student turns in a block 1
number of teacher words 42
number of student words in block 43
probability from DistilBERT: 0.1717
average, count
teacher said "turn" and "talk"? 0.0
number of student turns in a block 4.0
number of teacher words 36.5
number of student words in block 71.0
probability from DistilBERT: 0.35145
number of blocks 43
transcript feature set
ML regressor
2.126

Figure 4: Illustration of the transcript-level combined model on Indicator 3A, for a single PST utterance. Refer to Table 3 for model acronyms. The Indicator 3B model (RBX) proceeds analogously.

Snowball stemming, as implemented in NLTK (Loper and Bird, 2002), was used prior to computing word overlap. Each pair yields a 5-dimensional feature vector. The feature vector of a PST utterance is the mean-aggregate vector using the pairs in its block, skipping over utterances with fewer than 5 tokens. For PST utterances with no eligible student utterances in the block, we use the lowest possible value of the feature (e.g. 0 for JACCARD).

Combined classifiers (CAX and CBX) used both types of features. For Indicator 3A (CAX), the features were all the structure-only features (e.g. NUM_TEACHTOKS from CAS) as well as the DistilBERT-predicted positive probability (from CAN). Indicator 3B (CBX) followed analogously. Figure 3 is a cartoon summarizing which features appear in which classifier.

For structure-only classifiers (CAS and CBS) and combined classifiers (CAX and CBX), we used shallow learning models as implemented in the Scikit-learn toolkit (Buitinck et al., 2013). See Table 4 for the classifiers considered and Table 7 (in the Appendix) for the hyperparameter grid.

At the transcript level, Indicator regressor features are constructed by simple aggregates of utterance-level information. For content-only Indicator regressors (RAN and RBN), there are only two features: the relevant average DistilBERT-predicted probability (from CAN or CBN); and the count of PST utterances (or utterance blocks). For structure-only Indicator regressors (RAS and RBS), the features are the averages of the relevant structure-only features (e.g. NUM_TEACHTOKS from CAS, or JACCARD from CBS) and the count of PST utterances. For combined Indicator regressors (RAX and RBX), the features are the averages from both types of features and the count of PST utterances. Figure 4 in the Appendix is a cartoon summarizing how utterances are aggregated. See Table 4 for the regressors considered. See Table 8 (in the Appendix) for the hyperparameter grid.

As for Dimension 3 regressors (RDS, RDN, and RDX), features are simply the union of the features of the Indicator regressors. RDN inherits features from RAN and RBN; RDS inherits features from RAS and RBS; and RDX inherits features from RAX and RBX.

All experiments were carried out on a MacBook Pro laptop, with Apple M1 Pro chip. Computations did not use GPU.

4.3 Model selection and evaluation

For model selection, we performed a 5-fold cross-validation (CV) on the training set. Folds were split by PST, as described for the train/test partition (section 4.1).

For classifiers, the metric for model selection was κ (Cohen, 1960); higher values are better. For regressors, the metric was mean squared error (MSE); lower values are better. Since manual scores range from 1 to 3, the predicted score was truncated to this range. For each of the 15 models in Table 3, the final number of epochs (for LLMs) or final estimator (for shallow learning models) was selected using cross-validation in order to advance to test set evaluation. For choosing the numbers of epochs, the one-standard-error rule (Hastie et al., 2017) was used.

Models that used the predicted probability from DistilBERT as feature (i.e. all except CAS, CBS, RAS, RBS, and RDS) used the best-performing number of epochs from the corresponding fine-tuned LLM classifier.

5 Results

Table 5 shows the test set results for classifiers. For Indicator 3A, structure-based models dominated content-based models. For Indicator 3B, the trend was the opposite. For both Indicators, the combined models had the best performance. See Appendix for examples of positive-predicted utterances.

Table 6 shows the test set results for regressors. For Indicator 3A, structure and content models show similar performance. For Indicator 3B, the fine-tuned LLM dominated. For both Indicators, as well as the Dimension 3 score, the combined models showed the best performance.

6 Discussion

6.1 Modeling approach

Our results show that classifiers focused on the content of the PST utterance perform better for Indicator 3B, while those focused on the structure of the discourse perform better for Indicator 3A. Thus, the results suggest that it is quite difficult to get out of the *content* of a PST's utterance whether or not the utterance encouraged peer interaction. However, since the simulated students (a) do not tend to spontaneously engage in a multi-party discussion, yet (b) are compliant with the teacher's instructions, whether or not multiple students speak following the teacher is a fairly strong signal of whether the teacher encouraged them to do so.

In contrast, whether or not the teacher encouraged the students to engage with each others' ideas is easier to recover from the actual PST utterance than from evidence of lexical overlap or semantic similarity between subsequent student utterances. This may be because, given the highly constrained topic of the conversation (properties of the six powders), on the one hand, consecutive student utterances generally tend to have substantial textual overlap, whether or not the teacher encouraged that; on the other hand, overlap or semantic similarity as captured in pre-trained models may not be sufficiently nuanced to distinguish between actual uptake and mere accidental, topic-induced, semantic similarity or lexical overlap.

We observe that modeling the discourse dynamic explicitly and separately from the fine-tuned-LLM-based model of the content yields more explainable models than models where the content of a large surrounding context is used within the LLM-based model. Thus, our design and results allow us to see clearly the extent to which the fact of the within-block students' utterances, irrespective of what is said, can predict the score on Indicator 3A, as well as to observe the complementarity of the content and structure as sources of information.

6.2 Generalization based on select examples

We observed previously that the design of the human evaluation campaign conducted prior and independently from the computational modeling was such that raters were asked to provide justifications for their scores in the form of specific utterances that could serve as positive examples of the target behavior; only 5–8% of the PST utterances were picked as positive examples. The general prevalence of utterances that exhibit the target behavior was not known a-priori, nor was it obvious that better performance, based on holistic scores, would clearly correspond to having more utterances that exhibit such behaviors.

Figure 5 shows boxplots of the proportion of automatically predicted positive examples for either Indicator by human-assigned holistic proficiency levels according to Dimension 3 scores. First, we observe that the system was able to detect many more positive examples than were provided – even at the lowest level of performance, most PSTs exhibited the target behavior in more than 10% of their utterances, while most of the best-performing PSTs did it in more than 40% of theirs.

Second, we observe a strong differentiation between proficiency levels – boxes containing middle 50% of the performances per level have almost no overlap. This provides validity evidence not only for the automated modeling but for the human holistic scores as well, showing that they correspond to explicit, quantifiable transcript-level aggregation of relevant evidence.

Third, the emergent differentiation enables easily

Construct	Model	Number of epochs or estimator	Accuracy	Cohen κ	F1
Indicator 3A	CAN	3 epochs	0.899	0.283	0.321
	CAS	MP	0.924	0.604	0.646
	CAX	RF	0.931	0.641	0.679
Indicator 3B	CBN	6 epochs	0.774	0.531	0.711
	CBS	RF	0.632	0.260	0.602
	CBX	MP	0.793	0.571	0.739

Table 5: Test set evaluation results for utterance-level classifiers.

Construct	Model	Estimator	MSE	Pearson correlation
Indicator 3A	RAN	MP	0.343	0.468
	RAS	RF	0.354	0.480
	RAX	RF	0.335	0.513
Indicator 3B	RBN	LR	0.238	0.705
	RBS	MP	0.325	0.530
	RBX	LR	0.215	0.724
Dimension 3	RDN	BR	0.242	0.547
	RDS	MP	0.202	0.631
	RDX	BR	0.183	0.678

Table 6: Test set results for transcript-level regressors. Lower MSE is better; higher correlation is better.

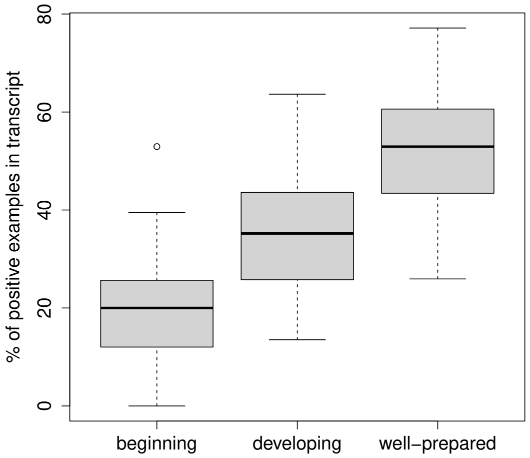


Figure 5: Boxplot of human-assigned Dimension 3 proficiency level vs. percentage of model-predicted positive examples (either Indicator) in transcript.

explainable and visually clear feedback whereby a PST's performance could be mapped against teachers at various levels of proficiency, to communicate not only current performance level but also how much more frequently one needs to implement the target behavior in order to move to the next level. Taken together, our results suggest that having humans provide select evidence for the score could be a viable alternative to a more comprehensive utterance-level annotation that is the prevalent approach in the literature on automation of the detailed evaluation of teacher discourse.

7 Conclusion

The goal of the current study was automated evaluation of teacher discourse when facilitating a discussion in a simulated elementary science classroom. We showed that models focused on the content of the teacher's utterances using fined-tuned large language models and models focused on the structure of the discourse modeled using handcrafted features captured complementary aspects of the target construct and could be fruitfully combined into hybrid models that outperformed both content and structure models. Our results also demonstrated strong generalization from a small number of "score justifications" provided by expert human raters, suggesting a potentially more efficient data generation paradigm than an exhaustive annotation of discourse moves.

8 Limitations

A limitation of the current study is the use of only one scenario for a simulated discussion, namely, the Mystery Powder task for an elementary science classroom and so it is not clear to what extent the type of models discussed in this paper will generalize to other scenarios. To address this limitation, we are developing additional scenarios, collecting discussion transcripts, and conducting human evaluations to generate data for additional studies that would examine the generalization of the technique proposed in this paper to new scenarios in both science and mathematics contexts.

Another limitation is that the current data come from pre-service teachers only; an online simulation could also be useful for early career in-service teachers. We are in the process of collecting data from in-service teachers and will be able to examine generalization to a different user population as the project progresses and more data become available for computational analysis.

Our experiments did not vary the size of the context window for DistilBERT. In line with Suresh et al. (2022b), it is possible that larger windows might substantially improve the performance of the fine-tuned-LLM-based models. That said, larger windows can potentially "encroach" on the structure-based models' territory making the distinction between what is due to the structure and what is due to the content harder to maintain, and with it, the explainability that comes from being able to point to the distinct aspects of the simulated discussions as information sources for the models. The explainability of the models is important not only for the PST buy-in, but also for the interdisciplinary team that is working on creating feedback reports based on the models' output. An explanation connecting the focus of the rubric to the performance of models with different types of information, as in section 6.1, helps the science teacher educators on the team appreciate the alignment between the rubric and the automated models.

Another limitation of the current study is using only DistilBERT. This model was picked for its efficiency and prior successful use in educational settings (Butt et al., 2022; Pearce et al., 2023); however, larger and more powerful models may support stronger performance, especially for Indicator 3A, where there is substantial room for improvement, with the current best performance of $r = 0.513$. Having established the baselines in this study, we intend to explore additional LLMs, resources permitting.

The data used in the study comes from predominantly White and female PSTs, reflecting the demographic at the data collection sites and in the teacher population in the USA. In the ongoing data collection, we are making an effort to reach out to more diverse demographics. Demographic information about the expert raters who provided scores and justifications was not collected; this will be rectified in future studies.

All current data come from pre-service teachers in the USA and all simulated discussions are conducted in English. In principle, actors who speak other languages could be trained to provide online practice to pre-service teachers in other cultural and linguistic environments; however, the detail and nuance of culturally appropriate teacher-student and student-student interactions might differ. At the moment, the scope and funding of the ongoing project do not allow addressing this limitation.

9 Ethics statement

The transcripts, scores and score justifications used in this study were collected with the approval of our Institutional Review Board with informed consent of the participants as part of previous studies. Participants were provided information about the purpose of the study, the risks and benefits to participating in the study, and details about what participation entailed. The raters were paid for the time they contributed to generating the scores and score justifications and the PSTs were paid for being research participants. The PSTs were enrolled in an elementary methods course at their university and were recruited based on their professor's participation in the study. Each PST could voluntarily consent to participate (or not) in the research study to have their transcript data used for research purposes. The consent form for participants included the following statement about risks: "Some participants may experience a small degree of discomfort when facilitating the discussions in the simulated classroom environments." All transcript data is deidentified, and a PST is represented by a numerical ID in each transcript. The data does not contain offensive content. The collected data is used in compliance with the consent. The consent form contained an explanation of the intended use: "The video recordings and transcripts of your sessions will be used for research purposes . . . anonymized data and recordings may be used in future research studies."

Since the ultimate goal of the project is to enable automated feedback to PSTs that would replace human feedback, there is a risk of incorrect feedback, since it is unlikely that an automated system will be accurate 100% of the time. First, human raters also sometimes make mistakes. Second, at least some of the use cases of the tool with feedback are within teacher training programs led by teacher educators; any feedback that surprised the PST or seemed unclear or incorrect can be discussed with the teacher educator. Third, every PST has access to the video recording of their own simulated discussion from Mursion; they can review the video to verify that the feedback makes sense with respect to their performance. Finally, a PST can engage in the simulation multiple times and it is possible that some of the feedback mistakes will be rectified in successive simulations.

Our use of the toolkits is in accordance with their licensing terms: Apache 2.0 license for HuggingFace transformers[3] and BSD 3.0 license for scikit-learn.[4]

References

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, Seattle, Washington. Association for Computational Linguistics.

Rhonda Bondie, Zid Mancenido, and Chris Dede. 2021. Interaction principles for digital puppeteering to promote teacher learning. *Journal of Research on Technology in Education*, 53(1):107–123.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122.

Ahmed Ashraf Butt, Saira Anwar, Ahmed Magooda, and Muhsin Menekse. 2022. Comparative analysis of the rule-based and machine learning approach for assessing student reflections. In *Porceedings of the 16th International Conference of the Learning Sciences – ICLS 2022*.

Courtney B. Cazden. 1988. *Classroom Discourse: The Language of Teaching and Learning*. Heinemann, Portsmouth, NH, USA.

Domenic V. Cicchetti. 1994. Guidelines, criteria, and rules of thumb for evaluating normed and standardized assessment instruments in psychology. *Psychological Assessment*, 6(4):284–290.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

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.

Tara Dalinger, Katherine B. Thomas, Susan Stansberry, and Ying Xu. 2020. A mixed reality simulation offers strategic practice for pre-service teachers. *Computers & Education*, 144(103696).

Debajyoti Datta, James P Bywater, Sarah Lilly, Jennifer L Chiu, Ginger S Watson, and Donald E Brown. 2023. Classifying mathematics teacher questions to support mathematical discourse. In *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium and Blue Sky. AIED 2023. Communications in Computer and Information Science*, volume 1831, Cham. Springer.

Dorottya Demszky and Heather Hill. 2023. The NCTE transcripts: A dataset of elementary math classroom transcripts. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 528–538, Toronto, Canada. Association for Computational Linguistics.

Dorottya Demszky and Jing Liu. 2023. M-Powering teachers: Natural language processing powered feedback improves 1:1 instruction and student outcomes. In *Proceedings of the Tenth ACM Conference on Learning @ Scale*, L@S '23, page 59–69, Copenhagen, Denmark. Association for Computing Machinery.

Dorottya Demszky, Jing Liu, Heather C. Hill, Dan Jurafsky, and Chris Piech. 2023. 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*.

Dorottya Demszky, Jing Liu, Zid Mancenido, Julie Cohen, Heather Hill, Dan Jurafsky, and Tatsunori Hashimoto. 2021. Measuring conversational uptake:

[3] https://github.com/huggingface/transformers/blob/main/LICENSE

[4] https://github.com/scikit-learn/scikit-learn/blob/main/COPYING

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.

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.

Lisa A. Dieker, Jacqueline A. Rodriguez, Benjamin Lignugaris/Kraft, Michael C. Hynes, and Charles E. Hughes. 2014. The potential of simulated environments in teacher education: Current and future possibilities. *Teacher Education and Special Education*, 37(1):21–33.

Zara Ersozlu, Susan Ledger, Alpay Ersozlu, Fiona Mayne, and Helen Wildy. 2021. Mixed-reality learning environments in teacher education: An analysis of TeachLivE™ research. *SAGE Open*, 11(3).

Evan J. Fishman, Hilda Borko, Jonathan Osborne, Florencia Gomez, Stephanie Rafanelli, Emily Reigh, Anita Tseng, Susan Million, and Eric Berson. 2017. A practice-based professional development program to support scientific argumentation from evidence in the elementary classroom. *Journal of Science Teacher Education*, 28(3):222–249.

GO Discuss Project. 2021. Scoring. Qualitative Data Repository.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2017. *Elements of Statistical Learning*, 2nd edition. Springer.

Annemarie H Hindman, Barbara A Wasik, and Kate Anderson. 2022. Using turn and talk to develop language: Observations in early classrooms. *Reading Teacher*, 76:6–13.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

Emily Jensen, Meghan Dale, Patrick J. Donnelly, Cathlyn Stone, Sean Kelly, Amanda Godley, and Sidney 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*, CHI '20, pages 1––13, Honolulu, HI, USA. Association for Computing Machinery.

Emily Jensen, Samuel L. Pugh, and Sidney K. D'Mello. 2021. A deep transfer learning approach to modeling teacher discourse in the classroom. In *LAK21: 11th International Learning Analytics and Knowledge Conference*, LAK21, page 302–312, Irvine, CA, USA. Association for Computing Machinery.

Vikram Kumaran, Jonathan Rowe, Bradford Mott, Snigdha Chaturvedi, and James Lester. 2023. Improving classroom dialogue act recognition from limited labeled data with self-supervised contrastive learning classifiers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10978–10992, Toronto, Canada. Association for Computational Linguistics.

Jay L. Lemke. 1990. *Talking Science: Language, Learning, and Values*. Ablex Publishing, Norwood, NJ, USA.

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.

Malinda Ann Hoskins Lloyd, Nancy J. Kolodziej, and Kathy Brashears. 2016. Classroom discourse: An essential component in building a classroom community. *School Community Journal*, 26(2):291–304.

Edward Loper and Steven Bird. 2002. NLTK: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pages 63–70, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Jamie N. Mikeska, Heather Howell, Joseph Ciofalo, Adam Devitt, Elizabeth Orlandi, Kenneth King, Michelle Lipari, and Glenn Simonelli. 2021. Conceptualization and development of a performance task for assessing and building elementary preservice teachers' ability to facilitate argumentation-focused discussions in science: The mystery powder task. ETS Research Memorandum RM-21-06.

Jamie N. Mikeska, Heather Howell, and Carrie Straub. 2019. Using performance tasks within simulated environments to assess teachers' ability to engage in coordinated, accumulated, and dynamic (CAD) competencies. *International Journal of Testing*, 19(2):128–147.

Tanya Nazaretsky, Jamie N Mikeska, and Beata Beigman Klebanov. 2023. Empowering teacher learning with AI: Automated evaluation of teacher attention to student ideas during argumentation-focused discussion. In *LAK23: 13th International Learning Analytics and Knowledge Conference*, page 122–132, Arlington, TX, USA. Association for Computing Machinery.

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.

Kate Pearce, Sharifa Alghowinem, and Cynthia Breazeal. 2023. Build-a-bot: Teaching conversational AI using a transformer-based intent recognition and question answering architecture. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 16025–16032.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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.

Victor Sampson and Margaret R. Blanchard. 2012. Science teachers and scientific argumentation: Trends in views and practice. *Journal of Research in Science Teaching*, 9:1122–1148.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108.

Jonathan T. Shemwell and Erin Marie Furtak. 2010. Science classroom discussion as scientific argumentation: A study of conceptually rich (and poor) student talk. *Educational Assessment*, 15(3-4):222–250.

Abhijit Suresh, Jennifer Jacobs, Charis Clevenger, Vivian Lai, James H Martin, and Tamara Sumner. 2021. Using AI to promote equitable classroom discussions: The TalkMoves application. In *AIED 2021: Artificial Intelligence in Education*, volume 12749 of *Lecture Notes in Computer Science*, Cham. Springer.

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.

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.

Harriet R. Tenenbaum, Naomi E. Winstone, Patrick J. Leman, and Rachel E. Avery. 2020. How effective is peer interaction in facilitating learning? A meta-analysis. *Journal of Educational Psychology*, 112(7):1303–1319.

Nhat Tran, Benjamin Pierce, Diane Litman, Richard Correnti, and Lindsay Clare Matsumura. 2023. Utilizing natural language processing for automated assessment of classroom discussion. In *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium and Blue Sky*, volume 1831 of *Computer and Information Science*, pages 490–496, Cham. Springer Nature Switzerland.

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.

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.

A Combined utterance-level classifier predictions

Here are some utterance blocks whose PST utterance was predicted positive for Indicator 3A, for the combined model (CAX).

TEACHER
Okay. I'm trying to figure out how to explain this the best way possible. Actually, Carlos, do you want to explain it because sometimes hearing it from a friend is easier.
CARLOS
Yeah. I didn't want to look at the weight because it just tells you how much there is, not what it is.
WILL
Hmm. Well that's a little confusing.
CARLOS
Well, what I mean is if you have a slice of pizza or you have a whole pizza, it's still pizza, right? It's just a different size.
WILL
Well, yeah, I guess so. A slice of pizza is still pizza.
CARLOS
Yeah, exactly and so it's the same with this. It doesn't matter how much you have, It's still the same thing.
WILL
I guess the weight doesn't actually tell you what it is.
CARLOS
Yeah, exactly.

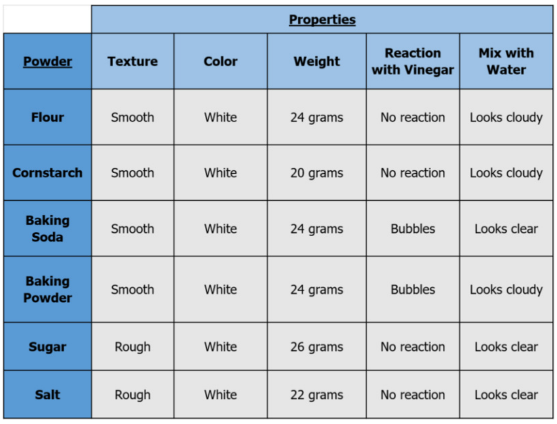
TEACHER
So in your small groups, and Carlos, you can join with Jayla and Emily. Talk about how you feel about the way that you went about the experiment and how you feel that you could've changed it.
EMILY
You know, I guess problem has enough properties, just not enough of the right properties.
CARLOS
Yeah. I thought I was on the right track with only using the weight, but I guess I didn't see it or realize that what the color in this is also the same one.
JAYLA
Yeah. I think because we were trying to be like "Lets test all the properties" but I guess now that we know that.

Here are some utterance blocks whose PST utterance was predicted positive for Indicator 3B, for the combined model (CBX).

TEACHER
Okay. Does anybody think that they should have looked at more properties or less? And why?
CARLOS
Well, I think that they should have looked at more properties because they only looked at a couple. And they were also talking about how they looked at weight and they didn't need to look at that one.

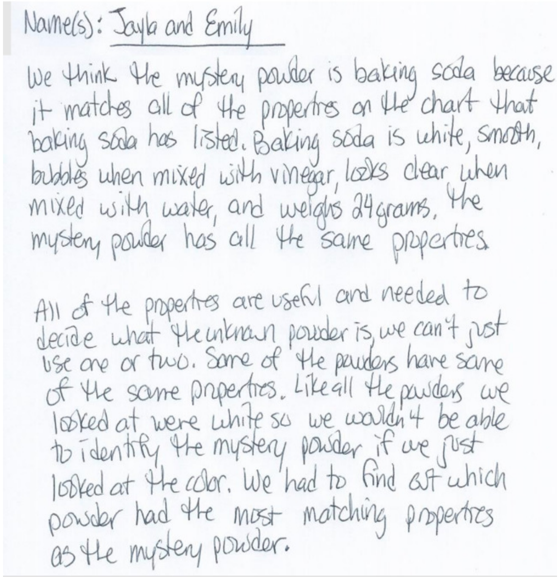
TEACHER
Okay. What does everyone else think about what Jayla just said?
WILL
Well, what she said about why weight is an important property I didn't think that. I thought it was an important property, because you could measure it. But what Carlos said makes sense.

B Some figures and tables



	Properties				
Powder	Texture	Color	Weight	Reaction with Vinegar	Mix with Water
Flour	Smooth	White	24 grams	No reaction	Looks cloudy
Cornstarch	Smooth	White	20 grams	No reaction	Looks cloudy
Baking Soda	Smooth	White	24 grams	Bubbles	Looks clear
Baking Powder	Smooth	White	24 grams	Bubbles	Looks cloudy
Sugar	Rough	White	26 grams	No reaction	Looks clear
Salt	Rough	White	22 grams	No reaction	Looks clear

Figure 6: Reference table of powders and properties for the Mystery Powder Task (Mikeska et al., 2021, p. 30).



Name(s): Jayla and Emily

We think the mystery powder is baking soda because it matches all of the propertres on the chart that baking soda has listed. Baking soda is white, smooth, bubbles when mixed with vinegar, looks clear when mixed with water, and weighs 24grams, the mystery powder has all the same propertres.

All of the propertres are useful and needed to decide what the unknown powder is, we can't just use one or two. Some of the powders have some of the same propertres. Like all the powders we looked at were white so we wouldn't be able to identify the mystery powder if we just looked at the color. We had to find out which powder had the most matching propertres as the mystery powder.

Figure 7: Pre-work by Jayla and Emily (Mikeska et al., 2021, p. 30).

```
cv_dict_classifier = {
"LR": (LogisticRegression(), {"C":[1,2,3,4,5,10,20], \
"class_weight":[None,"balanced"]}), \
"MLP": (MLPClassifier(random_state=42, max_iter=int(3e3)), {"hidden_layer_sizes": \
[1*(5,),2*(5,),3*(5,),1*(10,),2*(10,),3*(10,),1*(20,),2*(20,),3*(20,),1*(30,)
    ,2*(30,),3*(30,)], \
"activation": ["logistic", "tanh", "relu"], \
"solver": ["lbfgs", "sgd", "adam"], "alpha": [0.00005,0.0005]}), \
"DT": (DecisionTreeClassifier(random_state=42), { "splitter":["best","random"], \
"max_depth": np.arange(3, 15), "max_features":["log2","sqrt",None],"class_weight": [
    None,"balanced"]}),
"RF": (RandomForestClassifier(random_state=42), {"max_depth": [5,10,20,30,None],
"max_features": [1,"sqrt"],"min_samples_leaf": [1,2,4],"min_samples_split":
    [2,5,10],\
"class_weight": [None,"balanced"]})
}
```

Table 7: Classifier hyperparameter grids, for use with Scikit-learn.

```
cv_dict_regressor = {
"LR": (LinearRegression(), {"fit_intercept":[False, True]}), \
"MLP": (MLPRegressor(random_state=42,max_iter=int(3e3)), {"hidden_layer_sizes":
[1*(5,),2*(5,),3*(5,),1*(10,),2*(10,),3*(10,),1*(20,),2*(20,),3*(20,),1*(30,)
    ,2*(30,),3*(30,)],
"activation": ["logistic", "tanh", "relu"],
"solver": ["lbfgs", "sgd", "adam"], "alpha": [0.00005,0.0005]}), \
"DT": (DecisionTreeRegressor(random_state=42), { "splitter":["best","random"],
"max_depth": np.arange(3, 15), "max_features":["log2","sqrt",None]}), \
"BR": (BayesianRidge(), {"tol": [1e-4, 1e-3, 1e-2],
"alpha_1": [1e-7, 1e-6, 1e-5, 1e-4, 1e-3], "lambda_1": [1e-7, 1e-6, 1e-5, 1e-4, 1e
    -3],
"fit_intercept": [False, True]}), \
"RF": (RandomForestRegressor(random_state=42),{"max_depth": [5,10,20,30,None],
 "max_features": [1,"sqrt"], \
 "min_samples_leaf": [1,2,4], \
 "min_samples_split": [2,5,10]}),
}
```

Table 8: Regressor hyperparameter grids, for use with Scikit-learn.

Model	Selected	Cohen's κ mean (SE)
CAN	3 epochs	0.475 (0.001)
CAS	MP	0.653 (0.026)
CAX	RF	0.717 (0.026)
CBN	7 epochs	0.491 (0.007)
CBS	RF	0.324 (0.043)
CBX	MP	0.622 (0.033)

Table 9: 5-fold cross-validation results for classifiers, with Cohen's κ as metric. Higher values are better.

Model	Selected	MSE mean (SE)
RAN	MP	0.332 (0.046)
RAS	RF	0.250 (0.035)
RAX	RF	0.245 (0.042)
RBN	LR	0.242 (0.015)
RBS	MP	0.387 (0.017)
RBX	LR	0.236 (0.012)
RDN	BR	0.251 (0.044)
RDS	MP	0.233 (0.034)
RDX	BR	0.219 (0.034)

Table 10: 5-fold cross-validation results for regressors, with mean squared error (MSE) as metric. Lower values are better.

Explainable AI in Language Learning: Linking Empirical Evidence and Theoretical Concepts in Proficiency and Readability Modeling of Portuguese

Luisa Ribeiro-Flucht, Xiaobin Chen, Detmar Meurers
LEAD Graduate School and Research Network
University of Tübingen / Germany
luisa.ribeiro-flucht@uni-tuebingen.de
xiaobin.chen@uni-tuebingen.de
dm@sfs.uni-tuebingen.de

Abstract

While machine learning methods have supported significantly improved results in education research, a common deficiency lies in the explainability of the result. Explainable AI (XAI) aims to fill that gap by providing transparent, conceptually understandable explanations for the classification decisions, enhancing human comprehension and trust in the outcomes. This paper explores an XAI approach to proficiency and readability assessment employing a comprehensive set of 465 linguistic complexity measures. We identify theoretical descriptions associating such measures with varying levels of proficiency and readability and validate them using cross-corpus experiments employing supervised machine learning and Shapley Additive Explanations. The results not only highlight the utility of a diverse set of complexity measures in effectively modeling proficiency and readability in Portuguese, achieving a state-of-the-art accuracy of 0.70 in the proficiency classification task and of 0.84 in the readability classification task, but they largely corroborate the theoretical research assumptions, especially in the lexical domain.

1 Introduction

As technology evolves at a rapid pace, the field of education undergoes continuous adaptation. Particularly in language learning, numerous tools are being developed with the goal of facilitating the practice of a second language and providing tailored materials. In order to effectively model natural language, it's crucial to identify and empirically validate the relevant linguistic properties to use. Linguistic modelling with complexity measures has been proven to be highly effective in providing evidence-based insight into the assessment of both proficiency and readability (Benjamin, 2012; Crossley et al., 2017).

Second language proficiency and text readability are often associated concepts in language learning. Proficiency is usually equated to the notions of mastery and ability of understanding and producing another language (Hulstijn, 2015). Readability, in turn, encompasses the degree of reading difficulty which a text may exert on a reader (Dale and Chall, 1949). While widely acknowledged as multidimensional and dynamic constructs, proficiency and readability are commonly assessed using standardized scales. The Common European Framework of Reference for Languages (CEFR, Council of Europe) stands out as one of the most prominent scales for measuring proficiency, while readability is usually estimated according to different education, proficiency, and literacy levels.

In this context, it is essential to note the limited empirical evidence supporting the categorization of the mentioned constructs into levels and the precise definition of each level (Hulstijn, 2015). The English Profile Programme (EGP, Hawkins and Buttery, 2008) is a notable effort to clarify proficiency levels by identifying linguistic features whose presence or absence corresponds to specific English CEFR levels. However, the success of such an initiative heavily relies on the availability of abundant data and specialized manpower for data annotation and analysis, which may not be readily accessible for languages other than English.

In this paper, our objective is to propose an automatic method that comprehensively captures the nuanced characteristics defining language proficiency and text readability in Portuguese, as well as to provide a robust multilingual text analysis platform which will be made freely available online. By performing linguistic modeling and applying explanatory methods, we seek to validate theoretical postulations and enhance our understanding of the linguistic properties which are crucial for language learning, by answering the following research questions:

1. How well does a broad set of linguistic com-

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 199–209
June 20, 2024 ©2024 Association for Computational Linguistics

plexity measures model proficiency and readability levels in Portuguese?

2. What are the most discriminative measures for each proficiency and readability level? Do they coincide with theoretical suggestions?

3. Can Explainable AI be used with the purpose of describing proficiency and readability levels?

To address these questions, we use two distinct datasets: One consisting of European Portuguese learner productions and another consisting of Brazilian school materials. The constructs under investigation will be modeled as classification tasks. By leveraging such measures, language learning tools and generative models may better capture the nuances of written language, leading to a deeper understanding of the intricacies of readability and proficiency assessment (Housen et al., 2012).

2 Related Work

The extraction and analysis of linguistic complexity measures have been extensively explored in research. While many studies on automatic proficiency and readability assessment have primarily investigated the English language (e.g. Ortega, 2003; Lu, 2010; Bulté and Roothooft, 2020), advancements in Natural Language Processing (NLP) have facilitated the extension of research to other languages.

In European Portuguese, del Río (2019b) employed a supervised-learning approach using a learner corpus, achieving a 0.72 accuracy and 0.71 F-score by combining 39 linguistic complexity measures with other types of features, such as n-grams and readability formulas. Similarly, in Brazilian Portuguese, Evers (2013) used a corpus from a Brazilian Portuguese proficiency exam, extracting 48 linguistic measures for a binary classification task distinguishing between beginner and advanced learners, achieving an accuracy of 0.70 with a J48 classifier.

Automatic readability assessment has also been explored in the context of the Portuguese language. For instance, Curto et al. (2014) analyzed a corpus of L2 Portuguese texts, extracting 52 linguistic complexity measures. Their experiments achieved accuracy scores of 0.86 and 0.79 for three-level and five-level classification tasks, respectively. Additionally, Akef et al., 2024 extracted 489 linguistic complexity measures using the platform herein presented with machine learning algorithms for readability classification. This study demonstrated that models which incorporated informative features exhibited the highest generalization rate across various samples.

Regarding the use of explainable AI in Portuguese studies, Oliveira et al. (2023) explores the estimation of textual cohesion across essays in both Portuguese and English. The study found that although a deep learning-based model demonstrated superior performance, conventional machine learning models showed stronger potential in explainability.

The mentioned studies represent a crucial advancement in the automatic classification of proficiency and readability in Portuguese; however, they have limitations. Except for a few, most of the studies in Portuguese readability and proficiency assessment suffer from either a lack of a comprehensive set of measures, which might not fully capture the complexity and nuances of proficiency and readability, or from the absence of interpretability and detailed insight into feature importance. As a result, the depth of understanding regarding the constructs themselves and their categorization into separate levels may be limited.

3 Data

Two corpora were selected for our experiments and analyses: NLI-PT (Gayo et al., 2018) and Corpus de Complexidade Textual para Estágios Escolares do Sistema Educacional Brasileiro (Gazzola et al., 2019). The former comprises 3069 L2 Portuguese learner texts, categorized into three general levels: A (consisting of the CEFR levels A1 and A2), B (B1 and B2), and C (C1). The distributions of the texts in this corpus can be found in Table 1.

Proficiency Level	**Number of Texts**
A - Beginner	1,388
B - Intermediate	1,215
C - Advanced	466
Total	**3,069**

Table 1: Distribution of texts across proficiency levels in NLI-PT.

Regarding the latter corpus, it is a collection of 2076 Portuguese texts taken from Brazilian public school materials, and are separated into four school levels (elementary school, middle school, high school and university education). Their distribution is displayed in Table 2.

Education Level	Number of Texts
Elementary School	297
Middle School	325
High School	628
University Education	826
Total	**2,076**

Table 2: Distribution of texts across school levels in the Córpus de Complexidade Textual para Estágios Escolares do Sistema Educacional Brasileiro corpus.

It is important to note that the distribution of texts into the separate categories in both corpora is imbalanced. That means that some levels are better represented than others, possibly influencing the classification results.

4 Methods

Our experiments consist of three main steps: first, extracting linguistic complexity measures from the chosen corpora; then conducting two types of classification experiments, one for proficiency and one for readability; and finally, analyzing the results using an explainable AI method to understand feature importance.

4.1 Automatic Complexity Measure Extraction

The Common Text Analysis Platform (CTAP, Chen and Meurers, 2016),[1] which already supports other languages, was extended to accommodate the extraction of 465 Portuguese complexity measures covering superficial counts and the linguistic domains of lexicon, syntax, morphology and discourse.[2] Table 3 displays the current distribution of measures across these domains.

The extension to Portuguese analysis was made possible via the integration of the Stanza pipeline (Qi et al., 2020), which provides a pipeline for tokenization, lemmatization, sentence segmentation, part-of-speech tagging, morphological annotation, dependency and constituency parsing, followed by specific methods based on extraction rules and word frequencies. While the analysis tool is available online and is free to use, the Portuguese analysis feature is not yet online as of this publication.

The selection of which measures should be added to the Portuguese complexity measure set in this work was based on previous related works (Weiss and Meurers, 2019). Additionally, in order to understand which of these measures are associated with the different proficiency and readability levels, a detailed study was performed of the Camões Institute's Reference Level Descriptions (RLD, Referencial Camões, 2017; Vaz et al., 2019), which outlines the discursive notions, grammatical structures, and lexical items expected of learners based on their placement in the CEFR proficiency scale. The Manual for Syntactic Simplification for Portuguese (Specia et al., 2008) and the SIMPLEX-PB 3.0 database (Hartmann et al., 2018) were also consulted. These are resources which categorize vocabulary and linguistic structures as easy or difficult based on their occurrence in different readability levels.

Domain	Number of Measures
Superficial	26
Lexical	235
Syntactic	108
Morphological	52
Discourse-based	44
Total	**465**

Table 3: Distribution of linguistic complexity measures across the five domains.

4.1.1 Superficial Measures

Superficial aspects of the text are some of the most traditionally analyzed ones. Although these measures require minimum computational power, they have consistently shown good discriminative capabilities (Bulté and Roothooft, 2020; Housen et al., 2012; Norris and Ortega, 2009). These consist of linguistic element counts, lengths, normalizations and ratios.

The simplification manual suggests a uniform increase in the length of texts one can read as they advance in literacy. Regarding Portuguese as a second language, while the RLD does not explicitly mention an increase in superficial aspects of texts, it suggests that different noun and verb inflections as well as syntactic constituents are acquired as the learner moves to more advanced levels, which may consequently influence the length of their words, phrases and clauses.

[1] https://sifnos.sfs.uni-tuebingen.de/ctap/

[2] The complete list of measures can be found as a supplementary material.

4.1.2 Lexical Measures

Lexical complexity has been shown to be very relevant in linguistic complexity studies. For instance, Crossley et al. (2011) reported that older writers seem to produce more infrequent, less diverse, and more abstract words. It has been also demonstrated that the use of infrequent words may exert a negative impact on reading comprehension (Nation and Coady, 1988). In addition, McCarthy and Jarvis (2010) suggest that low values of lexical density may be indicative of a smaller propositional complexity. McNamara et al. (2010) also report on the positive correlation between lexical diversity and linguistic competence.

In this study, we extracted measures related to lexical density, variation, and sophistication. Lexical density was computed by scaling lexical and function words by the total number of tokens. Variation was assessed by dividing the number of lexical types by the number of lexical tokens, and through edit distance calculations for lemmas, parts of speech, and tokens. To measure lexical sophistication, we considered aspects such as age of acquisition (Cameirao and Vicente, 2010), concreteness, imageability, and familiarity (Soares et al., 2017). Contextual diversity and word frequencies were derived from the SUBTLEX-PT lexical database (Soares et al., 2015). Additionally, we included frequency-based measures from the Portuguese Vocabulary Profile project (Torigoe, 2017) and a list of complex words (Hartmann et al., 2018).

4.1.3 Morphological Measures

The morphological measures implemented in this work prioritize the inflectional system of Portuguese, as they can be easily generalized to other languages. Measures of derivational and compositional morphology will be eventually added to the system.

The RLD suggests that verb forms are learned incrementally starting from the simple present. The past participle verb form, for instance, is described as being incrementally learned, starting at A2 level, with its regular and irregular forms, and is consolidated at level B1 with the double participle with gender and number inflection. Given that this verb form is prevalent in constructions deemed as advanced, like passive sentences and the present perfect tense, it is expected to be more common in texts for advanced learners rather than beginners. Both the RLD and the simplification manual affirm that inflections like the present perfect tense, simple future tense, present subjunctive, conditional mood, and passive voice are typically found in advanced texts.

4.1.4 Syntactic Measures

The syntactic measures are based on both syntactic element counts and ratios. Clauses, phrases, complements, T-Units, modifiers, subjects and clefts were taken into consideration. We have measured clausal elaborateness, by taking clausal subordination and coordination into account. More specifically, regarding coordination, we calculate copulative, disjunctive and asyndetic coordinate clauses. In addition, we measure phrasal elaborateness by accounting for noun and verb phrases, as well as different types of subject, such as null and clausal subjects. Lastly, measures based on the Dependency Locality Theory (DLT, Gibson et al., 2000) were also included.

Most studies done in English have shown that measures like sentence length, clausal elaborateness, number of clauses and dependent clauses per T-unit increase throughout proficiency levels ((Norris and Ortega, 2009; Ortega, 2003; Lu, 2010). Moreover, the simplification manual suggests that sentences with a high rate of embeddedness are more challenging to read, as well as the inverse order of verb-subject, instead of subject-verb. The latter, being learned only at the B1 level, according to the RLD.

4.1.5 Discourse-based Measures

The measures implemented concerning discourse are based on the list of connectives developed by Mendes and Río (2018). Additionally, we measured the use of single and multi-word connectives, as well as easy and difficult connectives. The latter are based on two lists created by Leal et al. (2021). In terms of referential cohesion, measures regarding argument overlap, lemma overlap and lexical word overlap were calculated. For all these features, their mean and standard deviation values were also calculated and included as separate features.

In the simplification manual, it is suggested that discourse connectives improve comprehension, meaning they should occur often in earlier levels and be replaced incrementally by more advanced linguistic devices. The RLD also suggests that constructs like anaphora are acquired by intermediate learners.

4.2 Classification Experiments

Based on prior research findings (del Río, 2019a), we conducted classification tasks implementing three distinct supervised learning classifiers: Support Vector Machine, Linear Regression and Random Forest. In addition, a Multi-Layer Perceptron classifier was used in order to verify whether a simple neural network architecture performs significantly better or worse.[3]

For the sake of consistency and in order to warrant fair comparisons, all of the experiments herein performed were implemented in the Python programming language, using algorithms provided by the Scikit-learn library (Pedregosa et al., 2011).

The values for each measure were scaled using the library's method StandardScaler, in order to avoid the effect of high cardinality, due to differing range sizes among measures. Additionally, Scikit-learn's method GridSearch was applied alongside 10-fold cross-validation in order to optimize the models' performance and avoid overfitting. Lastly, to evaluate model performance, separate testing sets were created using an 80/20 split for training and testing purposes. Results from both the 10-fold cross-validation and held-out test sets are presented below.

4.3 Explainable Artificial Intelligence with SHAP

Explainable Artificial Intelligence has seen significant development, offering various approaches for understanding model outputs (Došilović et al., 2018). To gain insight into feature contributions, we adopted the Shapley Additive Explanations (SHAP, Lundberg and Lee, 2017) framework, which has been recently applied in proficiency and readability studies (e.g. Korniichuk and Boryczka, 2021; Nguyen and Wintner, 2022).

SHAP was selected over alternative interpretation methods such as LIME (Ribeiro et al., 2016) due to its ability to offer insights into feature importance both locally and globally, irrespective of the underlying model's complexity. This flexibility was crucial for our study, given the diverse linguistic measures and the use of non-linear SVMs with RBF kernels, where interpretation can be challenging (Sanz et al., 2018). In contrast, SHAP allows us to delve into each prediction, offering a deeper understanding of how specific features influence model outcomes.

Specifically, we employed the KernelExplainer method from the SHAP package. This method estimates the importance of each feature in making a particular prediction. It calculates the SHAP values, which represent the marginal contribution of each feature to the prediction across all possible combinations of features. Positive SHAP values indicate a feature's contribution to increasing the model's prediction, whereas negative values signify a decrease in the prediction. These values are then combined using a weighted sum to determine the overall importance of each feature.

5 Proficiency Classification Results

While all classifiers showed similar performance, the SVM classifier exhibited slightly better results compared to the others, as shown in Table 4. Conversely, the sole neural network architecture included in the analysis performed the worst. With 10-fold cross-validation, the best-performing classifier achieved a mean accuracy score of 0.70 and a weighted F-score of 0.68. Furthermore, on evaluation with the held-out test set, it achieved an accuracy of 0.73 and a weighted F-score of 0.72.

	10-Fold CV		Test Set	
	F1	Acc	F1	Acc
Logistic Regression	0.68	0.68	0.70	0.69
Multi Layer Perceptron	0.64	0.66	0.66	0.67
Random Forest	0.68	0.68	0.67	0.63
Support Vector Machine	0.68	0.70	0.73	0.72

Table 4: 10-fold cross-validation and test set accuracy and F1-scores achieved in proficiency classification experiments with all features.

In the confusion matrix (Table 5), proficiency level A had the highest accuracy, with 219 true positives, but 41 were misclassified as B. Notably, 51 texts from level B were misclassified as level A, and 19 as level C. Level C had the fewest true positives, possibly due to class imbalance, as discussed by (del Río, 2019b) or due to factor which have not been currently accounted for.

Figure 1 shows the mean SHAP values for the top 20 features with the most impact on the model's output for each proficiency level, listed in descending order. Among these features, 10 are related to the lexical domain. The most impactful feature is complex word frequency, particularly influential

[3] All experiment resources can be accessed through the following link: https://osf.io/ehdc9/?view$_{o}nly = 2e7ee278d187417c82219dc6eab6e29e$

	A	B	C
A	**219**	41	3
B	51	**178**	19
C	19	**32**	29

Table 5: Confusion matrix of the test set, obtained from the classification performed using the SVM classifier on all features.

for levels A and C. Additionally, two Portuguese Vocabulary Profile features, the A1 and B1 word lists, strongly influenced the prediction of level A.

Phrasal and clausal elaboration significantly influenced the model's output. The measures of relative clauses per clause and per T-unit were influential for distinguishing levels A and C, while the mean length of noun phrases is most impactful for predicting level B. The nominative case inflection emerges as the sole highly discriminative morphological measure. Additionally, word frequency-based features, clausal elaborateness, and lexical sophistication measures contribute to the list.

6 Readability Classification Results

Consistently with the proficiency classification experiments, Logistic Regression, Random Forest, Support Vector Machine and Multy-Layer Perceptron classifiers were implemented. The results achieved with 10-fold cross-validation and held-out test sets for each classifier are displayed in Table 6. Similarly to the proficiency experiments, the SVM classifier showed the best results, achieving an accuracy of 0.84 from 10-fold cross-validation, and an accuracy 0.85 with the held-out test set.

	10-Fold CV		Test Set	
	F1	Acc	F1	Acc
Logistic Regression	0.81	0.83	0.81	0.81
Multi Layer Perceptron	0.83	0.83	0.82	0.82
Random Forest	0.74	0.79	0.76	0.76
Support Vector Machine	0.85	0.86	0.87	0.87

Table 6: 10-fold cross-validation and test set accuracy and F1-scores achieved in the readability classification experiments with all features.

Upon reviewing the confusion matrix presented in Table 7, it becomes evident that the classifier effectively distinguished the elementary school school level from the others, with only 6 misclassifications as middle school texts. For the last three levels, there were minimal misclassifications into adjacent levels.

	1	2	3	4
1	**49**	6	0	0
2	4	**58**	12	1
3	0	1	**78**	12
4	0	1	14	**142**

Table 7: Confusion matrix of the test set, obtained from the classification performed using the SVM classifier on all features.

Figure 2 displays the mean SHAP values for the top twenty influential features. Thirteen of these features pertain to the lexical domain, four to morphology, two to surface features, and one to syntax.

The imageability of lexical word types had the strongest impact on the model's output. Familiarity, age of acquisition, the lexical density of articles and determiners and frequency-based measures were also highly discriminative. Additionally, superficial measures like the standard deviation of token length in syllables and letters were predictive. Morphological measures were also influential, with inflections in case, mood, person, and number showing strong impacts, particularly in differentiating the first and last levels. Notably, phrasal and clausal elaborateness seemed less significant in predicting school levels compared to proficiency classification.

7 Feature Selection

During the analysis of the measures, we found that some linguistic features were highly correlated with each other, aligning closely with expectations, for example, the correlation between the number of letters and the number of syllables, or the number of determiners and the number of articles. However, other correlations were less anticipated, such as those between subordinate clauses and corrected Type-Token Ratio (TTR) of verbs. Although removing correlated features is important for enhancing a model's performance, appreciating their interactions remains crucial for interpretation. Thus, a trade-off between interpretability, model complexity and performance emerges as a central consideration.

To maintain model interpretability, we refrained from employing feature engineering or dimensionality reduction techniques, opting instead for the

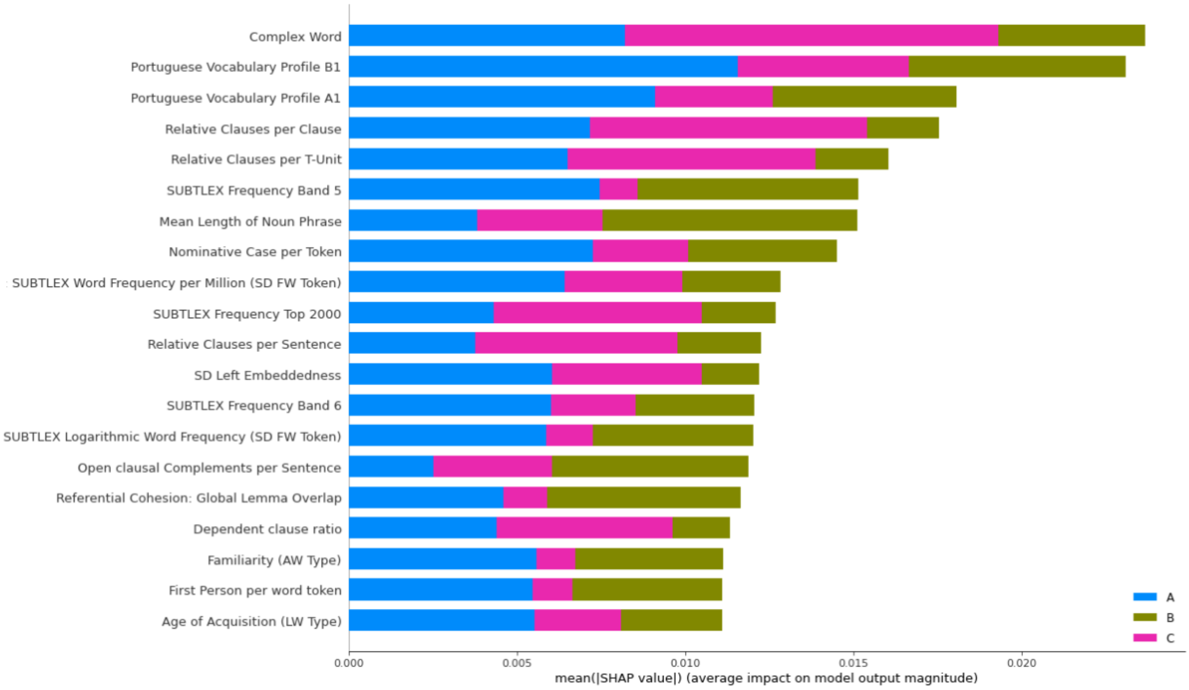


Figure 1: Distribution of the mean SHAP values of the 20 most discriminative features for proficiency level prediction (in descending order vertically). This figure also shows to what extent each measure, be its presence or absence, impacted the prediction of each level (on the horizontal axis).

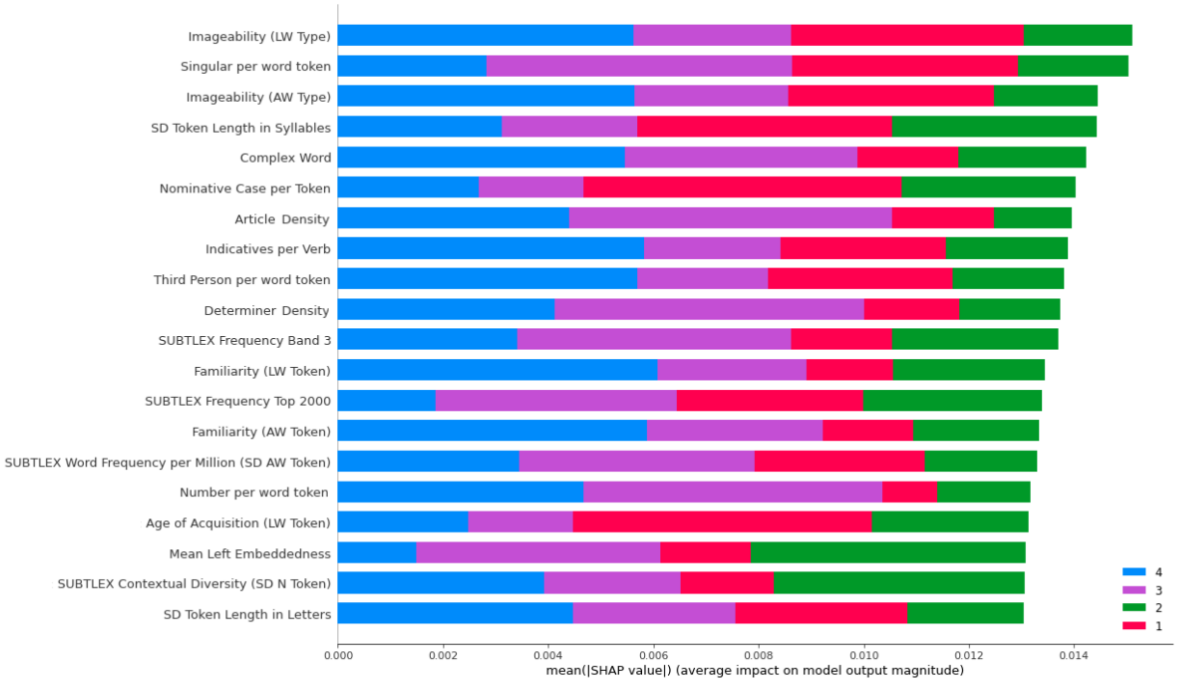


Figure 2: Distribution of the mean SHAP values of the 20 most discriminative features for readability level prediction.

CfsSubsetEval and InfoGain methods implemented by the Waikato Environment for Knowledge Analysis (WEKA, Hall et al., 2009). CfsSubsetEval identifies an informative yet uncorrelated subset of features. Similarly, InfoGain evaluates each feature's contribution to reducing entropy, aiding in the selection of the most informative features.

Reducing the number of features resulted in only a minor decline in performance, indicating that fewer features are adequate to achieve satisfactory classification results. Tables 8 and 9 provide a comprehensive overview of the SVM classifier's performance with both selected and full feature sets.

	10-Fold CV		Test Set	
	F1	Acc	F1	Acc
All features	0.68	0.70	0.73	0.72
CfsSubsetEval	0.65	0.67	0.68	0.68
InfoGain	0.66	0.65	0.68	0.67

Table 8: 10-fold cross-validation and test set accuracy and F1-scores achieved in the proficiency classification experiments with all the features and the selected feature sets.

	10-Fold CV		Test Set	
	F1	Acc	F1	Acc
All features	0.85	0.86	0.87	0.87
CfsSubsetEval	0.83	0.84	0.83	0.83
InfoGain	0.85	0.86	0.86	0.86

Table 9: 10-fold cross-validation and test set accuracy and F1-scores achieved in the readability classification experiments with all the features and the selected feature sets.

8 Discussion

In addition to obtaining the mean SHAP values of the most discriminative features, the SHAP values associated with each level were also inspected,[4] the measures referring to the superficial and lexical aspects exhibited the strongest discriminative power. Advanced learners produced more and longer words and sentences than beginners. A uniform increase was present concerning most of these features. The same is true for the school materials corpus herein utilized. Texts from the highest education levels demonstrated a higher incidence of words considered complex, abstract, infrequent and generally unfamiliar when compared to the lower ones. The same pattern was identified in terms of lexical variation. This is in line with the postulations in the simplification manual.

Regarding the syntactic domain, our data also corroborates most of the remarks. In the productions of L2 Portuguese learners, it was verified that clausal subjects, passive sentences, subordinate and relative clauses, as well as asyndetic coordinated clauses are indicative of more advanced levels. These grammatical constructions only arise after the general level B in the analyzed data. Comparatively, texts from the different educational levels demonstrated more homogeneity regarding syntactic measures. Although sentences and clauses are shorter in the early levels, constructions like subordinate and relative clauses as well as clausal subjects remained relatively constant across the levels. More pronounced contrasts regarding this domain were only found in terms of passive sentences and left embeddedness, which is in line with both the Portuguese RLD and the simplification manual.

Morphological measures also demonstrated contributions in differentiating the levels. For instance, it was observed that L2 learners placed at proficiency level A produced a distinctively higher amount of nominative case inflections, and, on the other hand, they exhibited low amounts of the accusative case inflection. In terms of verbal mood, it was observed that beginners also produce high amounts of indicative mood. This corroborates suggestions from the Portuguese RLD which suggests most verb tenses in the subjunctive mood are learned at levels B1 and B2. The same trend regarding the high use of the nominative case was observed in the Brazilian corpus. The elementary school texts contained a distinctively higher amount of this inflection when compared to high school or university ones; however the incidence of accusative case inflection was not as pronounced.

Concerning discursive measures, it has been suggested that as L2 learners progress, they tend to use fewer explicit cohesive devices (Crossley and McNamara, 2012). This trend was observed specifically for causal connectives: Their absence indi-

[4] The generated plots for each separate level can be found in the following link: https://osf.io/ehdc9/?view$_{o}nly = 2e7ee278d187417c82219dc6eab6e29e$

cated advanced L2 proficiency. On the other hand, regarding the school texts, this was the case for temporal connectives, which may very well point to a diminished presence of narrative discourse and higher amounts of expository discourse. Finally, in terms of referential cohesion, its low values were decisive for the prediction of the beginning proficiency level, but no impact was identified for the prediction of other levels or for the school texts.

9 Conclusion

In this paper, we explored Portuguese broad linguistic modeling in relation to L2 proficiency and text readability. Employing an elaborate NLP pipeline, we extracted 465 measures of linguistic complexity from two corpora. Our ultimate objective was to understand which measures exerted the most impact in each level's prediction and assess the extent to which these measures support the concept of holistic, static, ascending categories of proficiency and readability by implementing classification experiments and applying explainable AI methods.

Our results show that the consistent performance across different evaluation metrics suggests that the SVM classifier, trained on a broad set of linguistic complexity measures, provides a robust framework for modeling proficiency and readability levels in Portuguese texts. In particular, lexical features were found to have strong discriminative capabilities between different proficiency and readability levels. These findings provide evidence as to validate these measures and confirm the feasibility of modeling natural language using a diverse range of linguistic features. It also shows that XAI methods can be applied to linguistic complexity analysis.

In line with the Portuguese RLD and the simplification manual, the texts herein analyzed exhibited a uniform increase in the use of longer, more abstract, less familiar and less frequent words across both proficiency and readability levels. Moreover, an increase in sentence embeddedness and coordination, as well as tense and voice inflection was also positively confirmed in our findings. Additionally, trends in discursive measures suggest shifts in cohesive device usage as proficiency progresses, with possible implications for different discourse types.

These findings offer valuable insights for the refinement of language learning tools and assessment techniques. Specifically, they emphasize the significance of certain linguistic characteristics, such as vocabulary type, morphological and syntactic complexity, in modeling learner language and assessing proficiency and readability. Additionally, our intention to make CTAP's Portuguese analysis feature openly accessible online aims to support the development of more linguistically informed analyses through an accessible platform. This initiative is expected to facilitate the integration of linguistic insights into educational technologies.

Limitations

Although SHAP offers valuable insights, multicollinearity among highly correlated features may inflate or diminish feature importance, affecting SHAP interpretation. Despite potential changes in absolute SHAP magnitudes, relative importance rankings remain informative. SHAP values evaluate each feature's marginal contribution, taking into account feature interactions. Additionally, linguistic analyses lend credibility to SHAP interpretation.

The imbalance in both corpora underscores the necessity of balanced datasets to ensure reliable results in proficiency and readability assessment. An imbalanced corpus may lead to an overemphasis on dominant class characteristics, neglecting those of minority classes and affecting model performance. Another important observation is the influence that distinct topic and tasks may inflict in the emergence of specific grammar structures and lexical elements. These aspects have not been accounted for in these corpora's metadata, suggesting a need for future corpus creation that considers these aspects.

References

Soroosh Akef, Amália Mendes, Detmar Meurers, and Patrick Rebuschat. 2024. Investigating the generalizability of portuguese readability assessment models trained using linguistic complexity features. In *Proceedings of the 16th International Conference on Computational Processing of Portuguese*, pages 332–341.

Rebekah George Benjamin. 2012. Reconstructing readability: Recent developments and recommendations in the analysis of text difficulty. *Educational Psychology Review*, 24(1):63–88.

Bram Bulté and Hanne Roothooft. 2020. Investigating the interrelationship between rated l2 proficiency and linguistic complexity in l2 speech. *System*, 91:102246.

Manuela L Cameirao and Selene G Vicente. 2010. Age-of-acquisition norms for a set of 1,749 portuguese words. *Behavior research methods*, 42(2):474–480.

Xiaobin Chen and Detmar Meurers. 2016. CTAP: A web-based tool supporting automatic complexity analysis. In *Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity*, pages 113–119, Osaka, Japan. COLING.

Council of Europe. 2001. *Common European Framework of Reference for Languages: Learning, teaching, assessment*. Cambridge University Press, Cambridge.

Scott A Crossley and Danielle S McNamara. 2012. Predicting second language writing proficiency: The roles of cohesion and linguistic sophistication. *Journal of Research in Reading*, 35(2):115–135.

Scott A Crossley, Tom Salsbury, Danielle S McNamara, and Scott Jarvis. 2011. Predicting lexical proficiency in language learner texts using computational indices. *Language Testing*, 28(4):561–580.

Scott A Crossley, Stephen Skalicky, Mihai Dascalu, Danielle S McNamara, and Kristopher Kyle. 2017. Predicting text comprehension, processing, and familiarity in adult readers: New approaches to readability formulas. *Discourse Processes*, 54(5-6):340–359.

Pedro Curto, Nuno Mamede, and Jorge Baptista. 2014. Automatic readability classifier for european portuguese. *system*, 5:6.

Edgar Dale and Jeanne Chall. 1949. The concept of readability. *Elementary English*, 26(3).

Iria del Río. 2019a. Automatic proficiency classification in l2 portuguese. *Procesamiento del Lenguaje Natural*, 63:67–74.

Iria Iria del Río. 2019b. Linguistic features and proficiency classification in l2 spanish and l2 portuguese. In *Proceedings of the 8th Workshop on Natural Language Processing for Computer Assisted Language Learning (NLP4CALL 2019), September 30, Turku Finland*, 164, pages 31–40. Linköping University Electronic Press.

Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. 2018. Explainable artificial intelligence: A survey. In *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 0210–0215. IEEE.

Aline Evers. 2013. Processamento de língua natural e níveis de proficiência do português: um estudo de produções textuais do exame celpe-bras.

Iria Del Río Gayo, Marcos Zampieri, and Shervin Malmasi. 2018. A portuguese native language identification dataset. In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 291–296.

Murilo Gazzola, Sidney Evaldo Leal, and Sandra Maria Aluisio. 2019. Prediç ao da complexidade textual de recursos educacionais abertos em português.

Edward Gibson et al. 2000. The dependency locality theory: A distance-based theory of linguistic complexity. *Image, language, brain*, 2000:95–126.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

Nathan S Hartmann, Gustavo H Paetzold, and Sandra M Aluísio. 2018. Simplex-pb: A lexical simplification database and benchmark for portuguese. In *International Conference on Computational Processing of the Portuguese Language*, pages 272–283. Springer.

J Hawkins and Paula Buttery. 2008. Using learner language from corpora to profile levels of proficiency: Insights from the english profile programme. In *Language testing matters: Investigating the wider social and educational impact of assessment–proceedings of the ALTE Cambridge conference*, pages 158–175.

Alex Housen, Folkert Kuiken, and Ineke Vedder. 2012. *Dimensions of L2 performance and proficiency: Complexity, accuracy and fluency in SLA*, volume 32. John Benjamins Publishing.

Jan H Hulstijn. 2015. Language proficiency in native and non-native speakers. *Language Proficiency in Native and Non-native Speakers*, pages 1–206.

Ruslan Korniichuk and Mariusz Boryczka. 2021. Averaging and boosting methods in ensemble-based classifiers for text readability. *Procedia Computer Science*, 192:3677–3685.

Sidney Evaldo Leal, Magali Sanches Duran, Carolina Evaristo Scarton, Nathan Siegle Hartmann, and Sandra Maria Aluísio. 2021. Nilc-metrix: assessing the complexity of written and spoken language in brazilian portuguese. *arXiv preprint arXiv:2201.03445*.

Xiaofei Lu. 2010. Automatic analysis of syntactic complexity in second language writing. *International journal of corpus linguistics*, 15(4):474–496.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Philip M McCarthy and Scott Jarvis. 2010. Mtld, 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, Scott A Crossley, and Philip M McCarthy. 2010. Linguistic features of writing quality. *Written communication*, 27(1):57–86.

Amália Mendes and Iria del Río. 2018. Using a discourse bank and a lexicon for the automatic identification of discourse connectives. In *International Conference on Computational Processing of the Portuguese Language*, pages 211–221. Springer.

Paul Nation and James Coady. 1988. Vocabulary and reading. *Vocabulary and language teaching*, 97:110.

Isabelle Nguyen and Shuly Wintner. 2022. Predicting the proficiency level of nonnative hebrew authors. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5356–5365.

John M Norris and Lourdes Ortega. 2009. Towards an organic approach to investigating caf in instructed sla: The case of complexity. *Applied linguistics*, 30(4):555–578.

Hilário Oliveira, Rafael Ferreira Mello, Bruno Alexandre Barreiros Rosa, Mladen Rakovic, Pericles Miranda, Thiago Cordeiro, Seiji Isotani, Ig Bittencourt, and Dragan Gasevic. 2023. Towards explainable prediction of essay cohesion in portuguese and english. In *LAK23: 13th International Learning Analytics and Knowledge Conference*, pages 509–519.

Lourdes Ortega. 2003. Syntactic complexity measures and their relationship to l2 proficiency: A research synthesis of college-level l2 writing. *Applied linguistics*, 24(4):492–518.

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.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*.

PLE Referencial Camões. 2017. Direção de serviços de língua e cultura.

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.

Hector Sanz, Clarissa Valim, Esteban Vegas, Josep M Oller, and Ferran Reverter. 2018. Svm-rfe: selection and visualization of the most relevant features through non-linear kernels. *BMC bioinformatics*, 19:1–18.

Ana Paula Soares, Ana Santos Costa, João Machado, Montserrat Comesaña, and Helena Mendes Oliveira. 2017. The minho word pool: Norms for imageability, concreteness, and subjective frequency for 3,800 portuguese words. *Behavior Research Methods*, 49(3):1065–1081.

Ana Paula Soares, João Machado, Ana Costa, Álvaro Iriarte, Alberto Simões, José João de Almeida, Montserrat Comesaña, and Manuel Perea. 2015. On the advantages of word frequency and contextual diversity measures extracted from subtitles: The case of portuguese. *Quarterly Journal of Experimental Psychology*, 68(4):680–696.

Lucia Specia, Sandra Maria Aluísio, and Thiago A Salgueiro Pardo. 2008. Manual de simplificação sintática para o português. *Série de Relatórios do Núcleo Interinstitucional de Lingüística Computacional (NILC-TR-08-06)*.

Shintaro Torigoe. 2017. Portuguese vocabulary profile: uma lista de vocabulário a aprendentes do pl2/ple, baseada nos corpora de aprendentes e de livros de ensino. *Revista da Associação Portuguesa de Linguística*, (3):387–400.

Rui Vaz, Fátima Mendes, and Joana Batalha. 2019. Referencial camões ple. *VI Jornadas de Português Língua Estrangeira-Aquisição e Didática*.

Zarah Weiss and Detmar Meurers. 2019. Broad linguistic modeling is beneficial for german l2 proficiency assessment. In *Widening the Scope of Learner Corpus Research, Selected Papers from the Fourth Learner Corpus Research Conference, Louvain-la-Neuve: Presses Universitaires de Louvain*, pages 419–435.

Fairness in Automated Essay Scoring: A Comparative Analysis of Algorithms on German Learner Essays from Secondary Education

Nils-Jonathan Schaller[1], **Yuning Ding**[2], **Andrea Horbach**[2,3], **Jennifer Meyer**[1], **Thorben Jansen**[1]
[1]Leibniz Institute for Science and Mathematics Education at the University of Kiel, Germany
[2]CATALPA, FernUniversität in Hagen, Germany
[3]Hildesheim University, Germany

Abstract

Pursuing educational equity, particularly in writing instruction, requires that all students receive fair (i.e., accurate and unbiased) assessment and feedback on their texts. Automated Essay Scoring (AES) algorithms have so far focused on optimizing the mean accuracy of their scores and paid less attention to fair scores for all subgroups, although research shows that students receive unfair scores on their essays in relation to demographic variables, which in turn are related to their writing competence. We add to the literature arguing that AES should also optimize for fairness by presenting insights on the fairness of scoring algorithms on a corpus of learner texts in the German language and introduce the novelty of examining fairness on psychological and demographic differences in addition to demographic differences. We compare shallow learning, deep learning, and large language models with full and skewed subsets of training data to investigate what is needed for fair scoring. The results show that training on a skewed subset of higher and lower cognitive ability students shows no bias but very low accuracy for students outside the training set. Our results highlight the need for specific training data on all relevant user groups, not only for demographic background variables but also for cognitive abilities as psychological student characteristics.

1 Introduction

Educational equity is seen as a foundation for learning with technology (Warschauer et al., 2004), because all students need effective instruction. One of the most effective instructional practices is feedback (Hattie and Timperley, 2007), which can support students in acquiring complex skills like writing (Graham et al., 2015). Automated essay scoring (AES) can be used to provide students with feedback on their writing at scale (Fleckenstein et al., 2023).

The foundation of equity in automated feedback systems is the fairness of the algorithm ((Holstein and Doroudi, 2021), (Pedró et al., 2019)), i.e., the absence of any prejudice or favoritism toward groups of students based on their inherent or acquired characteristics, including their background and their psychological variables((Mehrabi et al., 2019),(Government Equalities Office, 2013)). Algorithmic fairness is widely discussed in various educational contexts from normative (Blodgett et al., 2020; European Commission, Directorate-General for Education, Youth, Sport and Culture, 2022), societal (Baker and Hawn, 2022; Kizilcec and Lee, 2020), or methodological (Mitchell et al., 2021) perspectives, but literature reviews have shown that it is rarely investigated empirically (Li et al., 2023). Specifically in the AES context, only six empirical studies have examined algorithmic fairness, examining differences in algorithmic accuracy and biases for students with different gender, race, and language backgrounds in English-language corpora (Arthur et al., 2021; Baffour et al., 2023; Bridgeman et al., 2009; Litman et al., 2021; Kwako et al., 2022; Yancey et al., 2023). This means that while AES is widely used in education in many countries (Fleckenstein et al., 2023) including non-English speaking countries, it is unclear whether the algorithms used are fair to all groups of students confronted with the results or whether they might disfavor some student gropus. Compounding the problem, the few existing studies have shown that, depending on the algorithms used, students' essays were not scored fairly and disfavored groups related to race/ethnicity, economic status, and English Language Learner status (e.g., Baffour et al. (2023); Litman et al. (2021); Yang et al. (2024)).

So far, previous studies only analyzed fairness in relation to students' demographic variables in corpora with students' essays in English: Extending this research to a corpus on argumentation essays in the German language, we address three main re-

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 210–221
June 20, 2024 ©2024 Association for Computational Linguistics

search questions: (1) How fair are AES algorithms for students with different levels of cognitive abilities as psychological characteristics strongly related to writing competence? (Zhang and Zhang, 2023). Addressing this question is linked to the wider equity issue of whether AES systems are likely to widen or narrow the gap between high and low-performing students. (2) How fair are AES algorithms in languages other than English? The question is especially important when automated scoring is based on large language models, mostly trained on English text data. (3) How is the distribution of student characteristics in the training data impacting the mean accuracy and fairness of the prediction?

By answering these questions, our paper makes the following contributions: First, we provide a set of baseline models, including shallow learning, deep learning, and generative large language models (LLM), for the newly released DARIUS corpus, thus enriching the automatic scoring landscape with models for a large German argumentative writing corpus.

Second, we conduct fairness evaluations on our results indicating that none of the models trained on the entirety of training data shows unfair behavior towards specific subgroups.

Finally, to assess the role of the distribution of the training data on algorithmic fairness, we train shallow and deep models with subsets of data from students of low and high cognitive ability, as well as a mixed subset based on low, medium and high cognitive ability, and show that the models are unfair to the groups not included in the training set.

We make all of our code publicly available.[1]

2 Related Work: Fairness in AES Algorithm

According to a literature review by Li et al. (2023), there have been 49 peer-reviewed empirical studies focused on fairness and predictive bias in education since 2010, highlighting the growing academic interest in these issues.

The studies included multiple fairness measures, including the accuracy for the included groups and the mean differences between predicted and annotated scores for each score (e.g., (Litman et al., 2021)). Most of these studies were conducted in contexts other than AES, such as predicting students' course performance or their likelihood of dropping out of a course. To our knowledge, there are only two papers that diagnosed the predictive bias displayed by AES models(Litman et al., 2021; Arthurs and Alvero, 2020), even though the importance of this task has been pointed out as early as in 2012 (Williamson et al., 2012). Litman et al. (2021) evaluated the fairness of shallow and deep learning AES algorithms for essays from the upper elementary level in the English language using three measures: Overall Score Accuracy (OSA), Overall Score Difference (OSD), and Conditional Score Difference (CSD). They found that shallow and deep AES algorithms showed systematically overly positive and negative scoring depending on students' gender, race, and socioeconomic status. Arthurs and Alvero (2020) showed that a shallow learning AES system for college admissions essays based on word vectors favored high-income students over low-income students (see also (Bridgeman et al., 2009) for similar results for essays from the Test of English as a foreign language). Additionally, the authors trained models on only essays from the highest quartile of students in terms of performance, showing that these models are not suitable for students from the other quartiles. Yang et al. (2024) further emphasized that the fairness of AES systems is compromised if such models are used on students or tasks for which they have not been trained.

In addition to the studies included in the literature review, recent studies added an investigation of fairness in Large Language Models scoring essays from a high school context Baffour et al. (2023) in the PERSUADE 2.0 corpus (Crossley et al., 2022). The authors compared the winning entries of the Kaggle Feedback Prize competition.[2] They show differences in the model's accuracy based on demographic factors such as student race/ethnicity, and economic disadvantage. Similar fairness issues based on students' demographic variables were shown for large language models in essays in the English language written by first (Kwako et al., 2023) and second language students (Yancey et al., 2023).

In summary, previous studies on fairness in AES have used shallow learning models, deep learning models, and LLMs and compared whether the accuracy of judgments and systematic over/underrating can be explained by students' demographic vari-

[1] https://github.com/darius-ipn/fairness_AES

[2] https://www.kaggle.com/competitions/feedback-prize-2021

ables. The results showed some fairness problems, which were exacerbated in the studies where the AES was additionally trained only on a homogenous group of students.

3 Data

The DARIUS corpus is a collection of 4,589 annotated argumentative texts written by 1,839 students from German high schools, spread across 114 classes in 33 different schools(Schaller et al., 2024). Essays that were off-topic, shorter than two sentences, empty, or contained names or other data relevant to data protection were removed beforehand. The final dataset consists of essays from two writing assignments focused on socio-scientific issues on the topics *energy* and *automotive*, containing 2,307 and 2,282 essays respectively. Students wrote a draft and revision on one task, followed by an essay on the other task, resulting in up to 3 essays per student. An example text is listed in the Appendix 7. Students also provided demographic data voluntarily, a selection of which is listed in Table 1.

The dataset has been extensively annotated with information about argumentative structure on different levels of granularity. In the present study, we focus specifically on a subset of these annotations, namely *content zone*, *major claim*, *position* and *warrant*. Out of the nine original annotation categories, we selected those as they reflect different parts of an argumentative text, e.g. structure and content, and are annotated on different granularity levels (token level to whole texts). We used the demographic data to measure fairness with respect to gender, profile, school, cognitive ability (KFT), and languages, which are further explained after providing more details on the annotations in Section 3.1. A more extensive description can be found in the original paper (Schaller et al., 2024).

3.1 Annotations

Content zone: This annotation category breaks down the essays into their basic parts: the introduction, the body, and the conclusion. Each section can be as short as one sentence or span several sentences.

Major claim annotation: Central to the argumentative essence of the essays, the Major Claim annotation identifies the pivotal stance taken by the author on the discussed issue. In contrast to similar annotation efforts (Stab and Gurevych, 2014), we also include claims written not only in the opening paragraphs but also within the conclusion, offering a comprehensive view of the argumentative intent. Such claims form the basis for the author's further arguments and the direction of their reasoning.

Position annotation: This annotation extracts the essay's directional stance regarding the thematic issues presented in the writing tasks — whether the argumentation aligns with, diverges from, or remains ambiguous towards the positions debated within the tasks. This annotation is important for understanding the diversity of viewpoints and the critical engagement of students with the socio-scientific topics at hand.

Warrant annotation: A warrant is one out of five argumentative elements annotated in the dataset as part of the Toulmin's Argumentation Pattern (TAP) annotations, following the definitions by Riemeier et al. (2012). TAP describes a structural framework for constructing logical and compelling arguments by including a claim, providing supporting evidence (data), explaining the connection between the claim and data (warrant), and addressing counterarguments (rebuttal). For this study, we focus exemplarily on warrants because the use of warrants indicates already a higher argumentation skill(Osborne et al., 2016). TAP elements are not marked on the sentence level but on the token level, as a TAP sequence can cover a wide range from subordinate clauses to entire paragraphs.

3.2 Demographic and Psychological Data

We consider the following demographic variables: **Grade** Grade indicates which grade level the student is in. The dataset was obtained for students between Grade 9 and Grade 12.

Gender The students could indicate their gender. Options were female, male, and diverse.

School The German school system differentiates between different forms of high school.

- Gemeinschaftsschule: non-academic track
- Gymnasium: academic track
- Berufsschule: vocational training

Profile The German school system allows students to choose a profile. The Natural Sciences profile, for example, has a focus on math and science, while the Social Sciences profile can have a focus on politics or ethics.

Languages The students could indicate the language that they speak at home.

Grade Level		Gender		Profile		Language	
Level	**Students**	**Gender**	**Students**	**Profile**	**Students**	**Language**	**Students**
9	423	Female	801	Natural Sciences	414	native	1265
10	346	Male	664	Social Sciences	255	non-native	576
11	547	Diverse	90	Sports	119		
12	404	Missing	284	Linguistics	61		
13	113			Aesthetics	13		
Missing	6			Missing	977		

Table 1: Combined Overview: Grade Level, Gender, Profile, and Language of Students

KFT The Cognitive Abilities Test (*Kognitiver Fähigkeitstest* or KFT) developed by Heller and Perleth (2000), measures students' cognitive abilities through non-verbal figural analogies. These questions evaluate abstract reasoning and the ability to apply logical rules to visual information without linguistic content, making them useful for assessing individuals across different linguistic backgrounds. A typical problem displays a sequence of shapes that follow a certain transformation (e.g., rotation, reflection). The test-taker must identify and apply the same transformation to a new set of figures.

4 Method

In the following section, we describe the experimental setup for our evaluation study.

4.1 Classifiers

We experiment with a diverse set of classifiers to see performance and fairness differences between instances of different model architectures. Our machine learning goal is to predict certain spans in an essay text. For most of these spans, span boundaries align with sentence boundaries.

Major claim annotations always consist of single sentences. The other annotation types, i.e. content zone and position annotations may also span multiple sentences. Only warrant annotations do not necessarily align with sentence boundaries and can consist of segments on the sub-sentence level. Therefore, we make use of both sentence classification and sequence tagging approaches. For sentence classification, we use a Support Vector Machine (SVM) in standard configuration, provided by the scikit-learn python package (Pedregosa et al., 2011) as an instance of shallow learning. The features utilized in the SVM classifier are the TF-IDF vectors of the most frequent 1- to 3-grams. We use a BERT-based [3] sentence classifier as an instance of deep learning and GPT-4 (OpenAI, 2024) to represent generative LLMs. For sequence tagging, we also use the BERT-based classifier and again prompt GPT-4 this time providing the whole essay as input.

4.2 Data Split

We use a fixed data split of 80% training data and 20 % test data. From the training data, we used a subset of 60% as validation data to find the best epoch for deep learning and for prompt-tuning for generative LLMs in pre-experiments, i.e. the whole training data set was used in the main experiments for training. As we were not interested in the overall best performance but rather in the intrinsic fairness differences between models, we did not further fine-tune any hyperparameters.

4.3 Performance and Fairness Evaluation

The evaluation of our classification results is motivated by the intended use of the classifiers to provide formative feedback to learners in e.g. an online tutoring system. Although it might also be of interest to show the specific location of an argumentative element within a learner essay as feedback, our primary concern for this study is to determine whether certain argumentative elements are present in a text or not. Therefore, we first transform any classifier output into a binary decision on the document level indicating whether (at least one instance of) a certain argumentative element is present in an essay.

In our fairness evaluation, we follow the framework proposed by (Loukina et al., 2019) and their implementation provided within the RSMTool software package (Madnani and Loukina, 2016). More precisely, we compute *overall score accuracy (osa)*, *overall score difference (osd)* and *conditional score difference (csd)*, where the first looks at squared errors $(S - H)^2$ and the latter two at actual errors $S - H$. In every case, a linear regression is fit with the error being the dependent variable and the

[3] dbmdz/bert-base-german-cased

Label	Model	All	Grades	Gender	Profile	School	Languages	KFT
Introduction	Shallow	.63	[.35, .68]	[.53, .67]	[.58, .73]	[.48, .68]	[.60, .70]	[.57, .67]
	Deep	.81	[.51, .85]	[.76, .84]	[.74, .83]	[.69, .95]	[.80, .85]	[.75, .85]
	LLM	.60	[.50, .63]	[.46, .62]	[.55, .61]	[.51, .77]	[.59, .59]	[.58, .61]
Conclusion	Shallow	.55	[.44, .71]	[.50, .58]	[.46, .55]	[.46, .61]	[.54, .55]	[.52, .57]
	Deep	.70	[.64, .80]	[.59, .74]	[.63, .81]	[.64, .78]	[.64, .71]	[.64, .78]
	LLM	.68	[.63, .76]	[.68, .81]	[.63, .67]	[.58, .84]	[.65, .68]	[.61, .72]
Major Claim	Shallow	.68	[.62, .74]	[.66, .74]	[.49, .75]	[.42, .81]	[.66, .72]	[.62, .72]
	Deep	.88	[.78, .92]	[.87, .88]	[.80, .95]	[.81, .89]	[.87, .88]	[.84, .90]
	LLM	.75	[.68, .82]	[.66, .81]	[.63, .84]	[.71, .91]	[.71, .86]	[.66, .86]
Position	Shallow	.41	[.34, .46]	[.34, .53]	[.16, .49]	[.29, .56]	[.36, .50]	[.17, .58]
	Deep	.44	[.23, .56]	[.36, .73]	[.23, .61]	[.28, .46]	[.37, .59]	[.27, .54]
	LLM	.32	[.13, .37]	[.29, .54]	[.29, .47]	[.22, .60]	[.31, .33]	[.23, .37]
Warrant	Shallow	.43	[.32, .51]	[.39, .51]	[.38, .51]	[.38, .47]	[.39, .55]	[.37, .52]
	Deep	.44	[.27, .53]	[.38, .55]	[.36, .68]	[.36, .65]	[.41, .52]	[.25, .54]
	LLM	.00	[-.16, .09]	[-.02, .32]	[-.18, .02]	[-.04, .14]	[-.02, .07]	[-.13, .08]

Table 2: Kappa values for the individual classifiers evaluated either on all test essays or on essays from a certain subgroup. We report the minimal and maximal values among the subgroups for each demographic variable.

respective subgroup information being the independent variable for osa and osd. For csd, two models are fitted, one with both the subgroup and human score as independent variables and one using the human score only. We use the R^2 as a measure of model fairness for osa and osd and the difference in R^2 for csd. In our analysis we follow Williamson et al. who established that absolute values above 0.1 suggests unfairness or bias against certain groups.

Fairness should be considered in addition to mean accuracy because research on teacher judgments has shown that the qualities of judgments are almost uncorrelated, and teachers who are very good at judging the average class level can be very unfair to the high or low-performing students((Möller et al., 2022),(Urhahne and Wijnia, 2021)).

We used Cohen's kappa to account for chance agreement in evaluating our model. This is crucial when classifiers evaluate argumentative elements in essays. Percentage agreement alone may overstate accuracy by reflecting chance, misleading results. Kappa provides a more accurate measurement of agreement strength. This is crucial in educational settings, where precise feedback is necessary, as ignoring chance agreement could overestimate teacher judgments. By incorporating kappa, we aim for a more balanced evaluation of our classifier's performance and fairness across diverse student groups, improving feedback in educational technologies and reducing biases in teacher assessments.

5 Experimental Study

In the following, we discuss the results of our experimental studies. We compare the three classification model types (**Shallow**, **Deep**, and **LLM**) with respect to both fairness and kappa. In the first experiment, we trained on the complete dataset and evaluated the fairness for certain subgroups.

In a second experiment, we trained models on subsets of the training data that represent only a specific part of the whole population (in our case, the upper and lower quartiles of the cognitive ability values) and examined the fairness of such models.

5.1 Evaluation of Full Models on Fairness and Performance

Table 2 presents the performance of our trained models with regard to chance-corrected kappa values, providing insights into the agreement between model predictions and human annotators. The range values in brackets show variances across the different subgroups. We excluded the subgroup Aesthetic from the category Profile, as it had only 9 students and led to extreme outliers. Our study involved three machine learning models: Shallow (SVM), Deep (BERT), and LLM (gpt-4-turbo-preview, GPT). The prompts used for the LLM are displayed in the Appendix.

For the prediction of the Introduction the Deep model demonstrated the highest performance with an overall kappa of .81, indicating a strong agreement with human annotations. In contrast, the Shallow and LLM models performed worse, a trend that persists through all models. The order of the model

Label	Metric	Model	Grades	Gender	Profile	School	Language	KFT
Introduction	osa	Shallow	.008	.001	-.002	-.001	.000	-.001
		Deep	.011	-.002	-.003	.001	-.001	.003
		LLM	-.000	-.001	-.004	.000	-.001	-.002
	osd	Shallow	.005	.004	-.001	.010	-.001	.007
		Deep	.001	.005	.000	-.001	-.001	-.000
		LLM	.014	-.001	.004	-.000	-.000	.001
	csd	Shallow	.019	.026	.038	.013	.001	.012
		Deep	.009	.022	.037	.004	.001	.000
		LLM	.032	-.002	.014	-.007	-.000	.008
Conclusion	osa	Shallow	.014	-.001	-.003	-.001	.000	.000
		Deep	-.003	.000	.007	-.002	-.001	.001
		LLM	.005	-.001	-.004	.002	-.001	-.002
	osd	Shallow	.004	.001	.004	.002	-.000	-.002
		Deep	-.002	-.001	.001	.006	.002	-.001
		LLM	.001	-.000	.000	.003	-.001	-.002
	csd	Shallow	-.003	.005	.019	-.001	.005	.005
		Deep	-.000	-.004	-.024	.004	-.001	-.002
		LLM	.003	-.007	.014	.000	-.000	-.000
Major Claim	osa	Shallow	-.002	-.002	-.004	.006	-.001	-.001
		Deep	.001	-.002	-.001	-.002	-.001	-.000
		LLM	-.001	.004	-.001	.001	.003	.005
	osd	Shallow	.003	-.001	-.002	.001	-.001	.007
		Deep	-.001	-.001	-.003	.000	-.001	-.000
		LLM	.004	-.002	-.002	-.002	.001	-.002
	csd	Shallow	.002	-.010	.011	.007	-.001	.005
		Deep	-.002	.001	.004	-.001	-.001	.000
		LLM	.002	.005	.044	.008	.003	-.001
Position	osa	Shallow	-.003	-.001	.003	.015	.001	.008
		Deep	.003	-.000	-.003	.017	-.001	.005
		LLM	.005	-.001	.004	.001	.001	.008
	osd	Shallow	.005	-.002	.012	.007	.002	.003
		Deep	-.000	-.001	-.002	.007	.001	-.002
		LLM	.004	-.002	.006	.007	-.001	.002
	csd	Shallow	.000	.012	.057	.019	.001	.010
		Deep	.002	.019	.050	.018	-.000	.014
		LLM	.008	-.010	-.018	-.005	.002	.022
Warrant	osa	Shallow	.007	-.002	.003	-.001	.007	.006
		Deep	.007	.001	.018	.008	.004	.016
		LLM	.012	.004	.005	-.003	.003	.008
	osd	Shallow	.000	.004	.002	.009	-.001	.003
		Deep	-.001	.002	-.002	-.001	-.001	-.000
		LLM	-.001	-.002	-.004	.004	-.000	.006
	csd	Shallow	.010	.002	-.036	.003	.000	-.001
		Deep	-.001	.011	-.008	.005	-.001	-.002
		LLM	.008	.006	.086	.007	.005	.025

Table 3: Fairness evaluation metrics of all classifiers.

performance is also reflected in the results ordered by demographic data.

For the Conclusion, the Deep model similarly outperformed its counterparts again, followed closely by the LLM. The SVM stays behind. When evaluating Major Claim, all models display a noticeably enhanced performance, especially the Deep model, reaching a kappa value of .88 followed by the LLM (.75), and lastly the Shallow model .68.

For Position and Warrant, kappa values reveal a drop in performance across all models, with the Deep model followed closely by the SVM. The LLM model lags behind, for the Position annotation at a value around zero, showing challenges in capturing the nuanced expression of stances or viewpoints within texts. Those results seem to mirror also the inter-annotator agreements of the original annotation, in which the annotations for Introduction/Conclusion (content zone) and Major Claim had both an inter-annotator Krippendorffs alpha of .83, the Position annotation at .68, while all TAP values (e.g. warrant) showed very low agreements.

The analysis reveals the strengths and weaknesses inherent to each modeling approach. Deep learning models, particularly BERT, consistently demonstrated robust kappa scores, affirming their suitability for complex linguistic tasks. Depending on the task, the SVM varied between staying behind between 1 to 18 points from BERT. In contrast, the generative capabilities of LLM models, such as GPT, varied extremely in their performance, although never outperforming the Deep model. These findings underscore the importance of model selection based on the specific demands of the task at hand. It is entirely possible that different prompts would have led to different results. However, it would have to be examined whether the resources required (time to develop and test the appropriate prompt, cost of the queries, energy consumption of LLM models) justify this procedure.

Table 3 shows the fairness measures based on the models, trained on the whole dataset. As reported, values over .10 are potentially an issue of concern. None of the calculations on any model resulted in any value above .10.

5.2 Training Models on KFT Subgroups

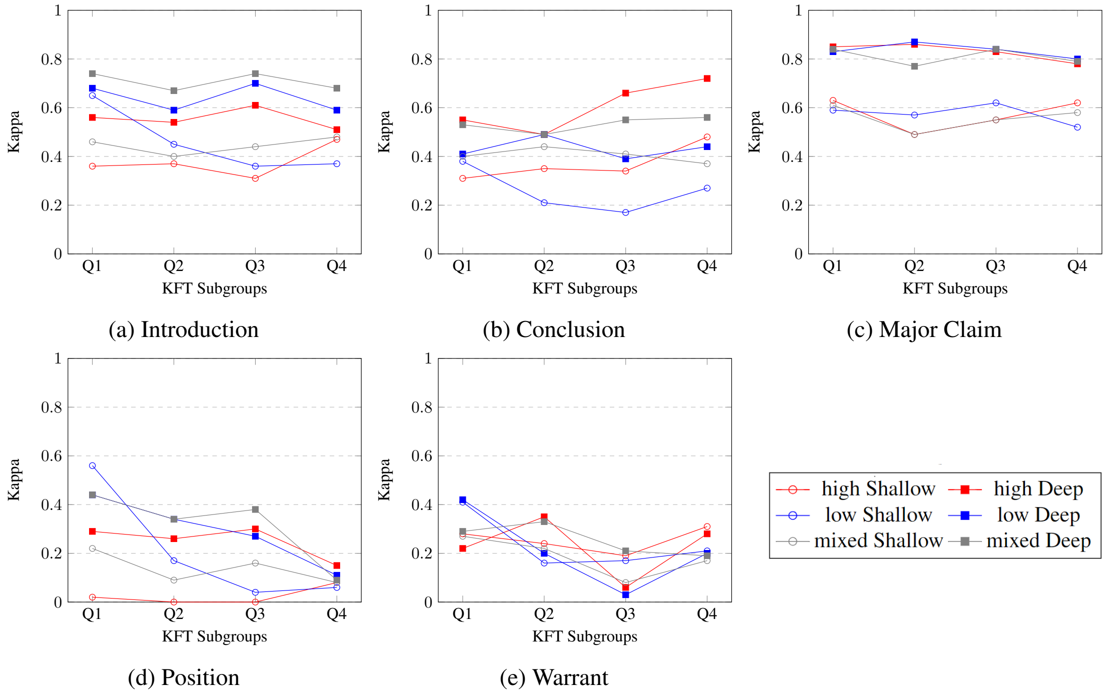
As a second step, we estimated the effects it can have if certain subgroups are not adequately reflected in the training data. For this experiment, we considered specifically cognitive abilities represented by cognitive ability values. We divided the training data into four quartiles based on the cognitive ability values and trained models on data from the lowest and highest quartiles only. For a more balanced comparison to general data, we also sampled a comparable size of training data from all four quartiles in a stratified way, e.g. from each quartile we took a randomised sample of 25%. This subset is further referret to as mixed data. This experiment was not conducted for LLMs, as our zero-shot approach does not rely on training data.

Unsurprisingly, the performance of both the SVM and the BERT model deteriorated in comparison to models trained on the full training set (see Table 4).

In general, the deep model performed still better than the shallow one, except for the position model trained on the low quartile as well as the warrant models trained on the highest and lowest quartiles. There is no indication that any of the quartiles lead to a stronger model. Each category

Label	KFT	Model	All	Grades	Gender	Profile	School	Languages
Introduction	high	Shallow	.38	[-.04, .44]	[.33, .62]	[.29, .39]	[.18, .48]	[.35, .45]
		Deep	.56	[.30, .62]	[.29, .59]	[.45, .56]	[.25, .57]	[.53, .61]
	low	Shallow	.47	[.26, .48]	[.30, .43]	[.30, .65]	[.41, .57]	[.40, .64]
		Deep	.65	[.59, .67]	[.63, .68]	[.60, .71]	[.62, .70]	[.64, .64]
	mixed	Shallow	.46	[.06, .51]	[.39, .61]	[.40, .47]	[.17, .55]	[.41, .57]
		Deep	.71	[.65, .73]	[.68, .71]	[.70, .76]	[.70, .73]	[.70, .75]
Conclusion	high	Shallow	.39	[.21, .48]	[.37, .53]	[.29, .47]	[.21, .52]	[.27, .40]
		Deep	.62	[.49, .66]	[.56, .65]	[.53, .72]	[.52, .77]	[.58, .62]
	low	Shallow	.25	[.19, .27]	[.17, .28]	[.21, .23]	[.09, .29]	[.20, .25]
		Deep	.44	[.16, .51]	[.40, .43]	[.29, .47]	[.34, .62]	[.41, .54]
	mixed	Shallow	.42	[.32, .55]	[.41, .56]	[.34, .44]	[.35, .45]	[.34, .42]
		Deep	.54	[.43, .57]	[.49, .69]	[.50, .63]	[.45, .62]	[.54, .54]
Major Claim	high	Shallow	.57	[.47, .63]	[.50, .62]	[.36, .58]	[.35, .57]	[.55, .61]
		Deep	.83	[.67, .87]	[.81, .88]	[.79, .90]	[.78, .92]	[.82, .85]
	low	Shallow	.58	[.46, .63]	[.52, .62]	[.37, .67]	[.35, .66]	[.57, .61]
		Deep	.84	[.77, .89]	[.80, .86]	[.76, .95]	[.70, .85]	[.82, .87]
	mixed	Shallow	.56	[.49, .62]	[.52, .56]	[.31, .70]	[.35, .58]	[.52, .67]
		Deep	.81	[.58, .86]	[.70, .82]	[.73, .89]	[.61, .82]	[.79, .87]
Position	high	Shallow	.02	[.00, .05]	[.00, .03]	[.00, .00]	[.00, .04]	[.00, .03]
		Deep	.29	[-.05, .43]	[.23, .49]	[-.04, .43]	[.17, .43]	[.27, .30]
	low	Shallow	.37	[.34, .48]	[.28, .69]	[.29, .41]	[.19, .69]	[.28, .52]
		Deep	.34	[-.07, .40]	[.28, .71]	[.23, .47]	[.08, .61]	[.29, .44]
	mixed	Shallow	.16	[.00, .18]	[.00, .15]	[.06, .15]	[.00, .37]	[.14, .18]
		Deep	.37	[-.03, .43]	[.33, .53]	[.24, .43]	[.29, .43]	[.33, .43]
Warrant	high	Shallow	.26	[.10, .32]	[.21, .29]	[.23, .29]	[.05, .27]	[.23, .36]
		Deep	.23	[.13, .30]	[.18, .31]	[.21, .34]	[.16, .37]	[.21, .29]
	low	Shallow	.23	[.19, .24]	[.19, .30]	[.20, .28]	[.14, .35]	[.19, .37]
		Deep	.20	[.03, .26]	[.16, .34]	[.19, .41]	[.12, .61]	[.16, .34]
	mixed	Shallow	.17	[.13, .22]	[.16, .28]	[.12, .31]	[.05, .41]	[.16, .22]
		Deep	.25	[.18, .30]	[.20, .39]	[.22, .28]	[.22, .49]	[.24, .29]

Table 4: Kappa values of KFT classifiers and all subtypes.



(a) Introduction

(b) Conclusion

(c) Major Claim

(d) Position

(e) Warrant

Figure 1: Kappa values of KFT classifiers on different KFT subgroups. Q = Quartile.

(low, high, and mixed) can perform best in different tasks, e.g. *mixed deep* in Introduction, *high deep* in Conclusion, or *low shallow/mixed deep* in Position. In terms of fairness, we still found no values above 0.1 (see Table 5).

Label	Metric	KFT	Model	Grades	Gender	Profile	School	Language	KFT
Introduction	osa	high	Shallow	.011	.005	-.004	.003	-.001	.001
			Deep	.001	.003	-.002	.000	-.001	.001
		low	Shallow	.001	.001	.007	.000	.008	.007
			Deep	-.003	-.001	-.003	-.002	-.001	.003
		mixed	Shallow	.008	-.001	-.004	.003	.001	.001
			Deep	-.003	-.000	-.004	-.003	-.001	.000
	osd	high	Shallow	.003	.003	-.004	.000	.001	-.000
			Deep	.001	-.001	.005	.000	-.000	.001
		low	Shallow	.002	.000	.003	.002	-.001	.006
			Deep	-.002	.007	.004	.002	-.001	.002
		mixed	Shallow	.010	-.002	.000	.009	-.001	.000
			Deep	-.001	.000	.002	.006	-.001	.004
	csd	high	Shallow	.012	.017	.093	.016	.000	-.001
			Deep	.014	.007	.074	.007	.007	.006
		low	Shallow	.018	.025	.053	.020	.005	.013
			Deep	.011	.008	.034	-.005	.002	.006
		mixed	Shallow	.022	.017	.065	.022	.002	.009
			Deep	.009	.011	.015	.004	-.000	.010
Conclusion	osa	high	Shallow	.011	.001	.003	.004	-.001	.002
			Deep	.000	-.000	.002	.001	-.001	.004
		low	Shallow	.010	-.000	-.004	.001	.002	.016
			Deep	.006	-.002	.000	-.001	.006	-.000
		mixed	Shallow	.011	.001	-.003	-.002	-.001	.001
		mixed	Deep	-.001	.003	-.001	-.001	-.001	-.003
	osd	high	Shallow	.016	-.002	.005	.004	-.001	-.001
			Deep	-.004	.002	-.004	-.003	-.000	-.001
		low	Shallow	.004	-.002	-.002	.000	.003	.012
			Deep	.005	-.002	.001	-.000	-.001	-.003
		mixed	Shallow	.003	-.002	-.000	.001	-.001	-.003
			Deep	-.001	-.001	.003	-.002	-.001	-.002
	csd	high	Shallow	.010	-.009	-.025	-.007	.006	.010
			Deep	.004	.003	-.007	-.003	-.001	.004
		low	Shallow	.001	.006	-.033	.006	-.000	-.001
			Deep	.004	.012	.042	.008	.001	.002
		mixed	Shallow	.001	-.009	-.003	-.011	.004	.003
			Deep	.004	.006	.034	.004	.001	.007
Major Claim	osa	high	Shallow	-.001	.004	-.004	.008	-.001	.000
			Deep	.001	-.002	-.003	.004	-.001	-.003
		low	Shallow	-.002	.003	-.001	.003	-.001	-.003
			Deep	.001	-.000	.002	-.001	-.000	-.002
		mixed	Shallow	.000	-.001	-.001	.018	.000	-.001
			Deep	.006	.001	.003	.003	.001	-.002
	osd	high	Shallow	-.001	.000	-.002	-.003	-.000	.005
			Deep	-.002	-.000	-.004	-.003	-.000	-.002
		low	Shallow	.004	.002	.002	.000	-.001	.000
			Deep	.003	-.000	-.004	-.002	.000	-.000
		mixed	Shallow	.006	-.002	-.002	-.003	-.001	.003
			Deep	.002	-.001	-.001	-.001	-.001	-.001
	csd	high	Shallow	-.002	-.004	.032	.014	-.001	.005
			Deep	-.002	.006	-.010	.005	.001	-.002
		low	Shallow	.002	.002	.043	.014	-.001	-.000
			Deep	.002	-.001	-.003	-.005	-.000	-.000
		mixed	Shallow	.005	.000	.020	.021	-.000	.004
			Deep	.002	.000	.012	.004	-.001	-.001
Position	osa	high	Shallow	.003	.002	.020	.011	.014	.036
			Deep	-.002	-.002	-.002	.012	.007	.024
		low	Shallow	.002	-.000	.007	.016	.000	.015
			Deep	-.001	-.001	-.000	.003	.001	.005
		mixed	Shallow	.001	.003	.016	.008	.009	.026
			Deep	-.001	-.002	.020	.009	.002	.011
	osd	high	Shallow	.003	.002	.020	.011	.014	.036
			Deep	.000	-.000	-.001	.006	.001	.003
		low	Shallow	.000	-.002	.005	.006	-.001	.006
			Deep	-.003	-.002	-.001	.005	.000	.003
		mixed	Shallow	.002	.003	.017	.006	.010	.024
			Deep	-.002	-.002	.005	.001	.003	.002
	csd	high	Shallow	-.000	-.003	.015	-.003	.000	-.000
			Deep	.003	-.013	.096	-.014	.001	.004
		low	Shallow	-.001	.030	.027	.039	.005	.013
			Deep	.001	.008	.017	.016	.001	.003
		mixed	Shallow	-.001	.019	.041	.025	-.000	.001
			Deep	.002	.005	.059	.005	-.000	.004
Warrant	osa	high	Shallow	.010	-.001	-.000	-.002	.006	.004
			Deep	.003	-.000	.003	.007	.004	.015
		low	Shallow	.014	-.002	.001	-.000	.008	.010
			Deep	.011	-.001	.012	.013	.008	.023
		mixed	Shallow	.019	-.002	.013	.007	.004	.015
			Deep	.007	.001	.002	.001	.003	.009
	osd	high	Shallow	.005	-.002	-.002	.004	-.001	.001
			Deep	.000	-.000	-.003	-.002	-.001	-.001
		low	Shallow	.003	-.001	.016	.011	.001	.013
			Deep	.005	-.002	.002	.001	.000	.001
		mixed	Shallow	.012	-.002	.009	.011	-.000	.007
			Deep	.005	-.002	-.002	-.001	-.001	.002
	csd	high	Shallow	.002	-.003	-.047	-.002	.000	.003
			Deep	.009	.007	-.020	.002	.002	-.000
		low	Shallow	.003	-.008	-.042	-.001	-.000	.003
			Deep	-.000	-.005	-.035	-.002	-.001	-.001
		mixed	Shallow	.001	-.017	-.045	-.007	.000	-.001
			Deep	.000	-.011	-.014	-.008	.002	.001

Table 5: Fairness evaluation metrics of KFT classifiers and all subtypes.

When examining Figure 1 we can see that models differed in their performance when tested on different subgroups. For the Introduction, a shallow model trained on the dataset of the students with the highest KFT quartile (*high shallow*) was performing better on the subgroup it was trained on (e.g. Quartile 4) than on the other subgroups and the other way around (low KFT model performed better on the subset with low KFT, e.g. Quartile 1.). The mixed models had the lowest variance in performance.

There are exceptions in which the model performed better on a different subgroup than the one it was trained on, e.g., in (d) Position, all models except high shallow lost performance on Quartile 4. Furthermore, all combinations of algorithm and training data did have a comparable stable performance on (c) Major Claim.

In general, using training data from only one student group seemed to introduce a bias, disadvantaging other student groups. This finding underlines the need to include training data from a diverse range of students to ensure fairness and avoid skewed outcomes.

6 Conclusion and Future Work

In our work, we provide three basic models (shallow learning models, deep learning models, and LLM) trained on the annotations of the DARIUS corpus of learner texts in German. These models are ready to use in schools, for example, to create a feedback tool for training argumentative skills. Evaluation of model fairness showed that all models produced fair scores for all students, considering demographic and psychological differences among students. In a second experiment, we trained our models on subgroups of students, based on either low, high, or mixed cognitive abilities, to investigate the extent to which skewed training data leads to unfair AES system scores. Our results showed lower performance for students who were not in the training data, emphasizing the importance of including samples of the full range of users in the training data for AES, not only for demographic background variables but also for psychological aspects such as cognitive abilites. Failure to do so risks reducing the predictive accuracy of the algorithm for those who are not adequately represented. To mitigate the risk of students receiving unfair scores based on their demographic and psychological variables, we advocate that future AES systems incorporate the goal of fairness in addition to accuracy into their training data collection and algorithm optimization function, going beyond the current state of retrospective analysis of model fairness.

7 Limitations

This study encounters several limitations that have to be mentioned. One constraint is the small size of certain subgroups within the corpus, as seen in Table 1, e.g., students with specific family languages, profiles like Linguistics or Aesthetics. The underrepresentation of those subgroups poses a challenge in drawing robust conclusions for these particular groups, potentially impacting the reliability and applicability of our outcomes to these populations.

Additionally, the comparatively homogenous population in the state of Schleswig-Holstein in northern Germany, restricts the generalizability of our findings. The demographic profile of Schleswig-Holstein may not reflect the diversity found in other regions or countries, potentially narrowing our study's insights.

In conclusion, while our study provides insights into fairness in the subgroups of the DARIUS Corpus, these limitations underscore the necessity for a cautious interpretation of our findings and suggest areas for future research efforts to build upon and address these constraints.

8 Acknowledgements

This work was supported by the Deutsche Telekom Stiftung and partially conducted at "CATALPA - Center of Advanced Technology for Assisted Learning and Predictive Analytics" of the FernUniversität in Hagen, Germany.

References

Philip Arthur, Dongwon Ryu, and Gholamreza Haffari. 2021. Multilingual simultaneous neural machine translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4758–4766, Online. Association for Computational Linguistics.

Noah Arthurs and AJ Alvero. 2020. Whose truth is the "ground truth"? college admissions essays and bias in word vector evaluation methods. *International Educational Data Mining Society*.

Perpetual Baffour, Tor Saxberg, and Scott Crossley. 2023. Analyzing bias in large language model solutions for assisted writing feedback tools: Lessons from the feedback prize competition series. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 242–246.

Ryan S. Baker and Aaron. Hawn. 2022. Algorithmic bias in education. *International Journal of Artificial Intelligence in Education*, 32:1052–1092.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of "bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online. Association for Computational Linguistics.

Brent Bridgeman, Catherine Trapani, and Yigal Attali. 2009. Considering fairness and validity in evaluating automated scoring. In *Annual Meeting of the National Council on Measurement in Education*, San Diego, CA.

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 (PERSUADE) corpus 1.0. *Assessing Writing*, 54:100667.

European Commission, Directorate-General for Education, Youth, Sport and Culture. 2022. *Ethical guidelines on the use of artificial intelligence (AI) and data in teaching and learning for educators*. Publications Office of the European Union.

Johanna Fleckenstein, Lucas W. Liebenow, and Jennifer Meyer. 2023. Automated feedback and writing: A multi-level meta-analysis of effects on students' performance. *Frontiers in Artificial Intelligence*, 6.

Government Equalities Office. 2013. Equality Act 2010: guidance. `https://www.gov.uk/guidance/equality-act-2010-guidance`. Accessed: 2023-09-21.

Steve Graham, Michael Hebert, and Karen Harris. 2015. Formative assessment and writing: A meta-analysis. *The Elementary School Journal*.

John Hattie and Helen Timperley. 2007. The power of feedback. *Review of Educational Research*.

Kurt Heller and Christoph Perleth. 2000. *Kognitiver Fähigkeitstest für 4.-12. Klassen, Revision (KFT 4-12+ R)*.

Kenneth Holstein and Shayan Doroudi. 2021. Equity and artificial intelligence in education: Will "aied" amplify or alleviate inequities in education? *CoRR*, abs/2104.12920.

René F. Kizilcec and Hansol Lee. 2020. Algorithmic fairness in education. *CoRR*, abs/2007.05443.

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, Seattle, Washington. Association for Computational Linguistics.

Alexander Kwako, Yixin Wan, Jieyu Zhao, Mark Hansen, Kai-Wei Chang, and Li Cai. 2023. Does bert exacerbate gender or l1 biases in automated english speaking assessment? In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 668–681.

Lin Li, Lele Sha, Yuheng Li, Mladen Raković, Jia Rong, Srecko Joksimovic, Neil Selwyn, Dragan Gašević, and Guanliang Chen. 2023. Moral machines or tyranny of the majority? a systematic review on predictive bias in education. In *LAK23: 13th International Learning Analytics and Knowledge Conference*, pages 499–508.

Diane Litman, Haoran Zhang, Richard Correnti, Lindsay Matsumura, and Elaine L. Wang. 2021. A fairness evaluation of automated methods for scoring text evidence usage in writing. In *International Conference on Artificial Intelligence in Education*, pages 255–267, Cham. Springer International Publishing.

Anastassia Loukina, Nitin Madnani, and Klaus Zechner. 2019. The many dimensions of algorithmic fairness in educational applications. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–10.

Nitin Madnani and Anastassia Loukina. 2016. RSMTool: A collection of tools for building and evaluating automated scoring models. *Journal of Open Source Software*, 1(3).

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2019. A survey on bias and fairness in machine learning. *ACM Computing Surveys*.

Shira Mitchell, Eric Potash, Solon Barocas, Alexander D'Amour, and Kristian Lum. 2021. Algorithmic fairness: Choices, assumptions, and definitions. *Annual Review of Statistics and its Application*.

Jens Möller, Thorben Jansen, Johanna Fleckenstein, Nils Machts, Jennifer Meyer, and Raja Reble. 2022. Judgment accuracy of german student texts: Do teacher experience and content knowledge matter? *Teaching and Teacher Education*, 119:103879.

OpenAI. 2024. Gpt-4 technical report.

Jonathan Osborne, Bryan Henderson, Anna Macpherson, Evan Szu, Andrew Wild, and Shi-Ying Yao. 2016. The development and validation of a learning progression for argumentation in science. *Journal of Research in Science Teaching*, 53.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Andreas Müller, Joel Nothman, Gilles Louppe, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Francesc Pedró, Miguel Subosa, Axel Rivas, and Paula Valverde. 2019. Artificial intelligence in education: challenges and opportunities for sustainable development. Working papers on education policy 7, UNESCO, France. Includes bibliography.

Tanja Riemeier, Claudia Aufschnaiter, Jan Fleischhauer, and Christian Rogge. 2012. Argumentationen von schülern prozessbasiert analysieren: Ansatz, vorgehen, befunde und implikationen. *Zeitschrift für Didaktik der Naturwissenschaften*, 18:141–180.

Nils-Jonathan Schaller, Andrea Horbach, Lars Höft, Yuning Ding, Jan L Bahr, Jennifer Meyer, and Thorben Jansen. 2024. Darius: A comprehensive learner corpus for argument mining in german-language essays. OSF Preprints. Accepted for LREC-COLING 2024.

Christian Stab and Iryna Gurevych. 2014. Annotating argument components and relations in persuasive essays. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1501–1510, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Detlef Urhahne and Lisette Wijnia. 2021. A review on the accuracy of teacher judgments. *Educational Research Review*, 32.

Mark Warschauer, Michele Knobel, and Leeann Stone. 2004. Technology and equity in schooling: Deconstructing the digital divide. *Educational Policy*.

David M. Williamson, Xiaoming Xi, and F. Jay Breyer. 2012. A framework for evaluation and use of automated scoring. *Educational Measurement: Issues and Practice*, 31.

Kevin P. Yancey, Geoffrey T. LaFlair, Anthony Verardi, and Jill Burstein. 2023. Rating short l2 essays on the cefr scale with gpt-4. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications, BEA@ACL 2023, Toronto, Canada, 13 July 2023*, pages 576–584. Association for Computational Linguistics.

Kaixun Yang, Mladen Raković, Yuyang Li, Quanlong Guan, Dragan Gašević, and Guanliang Chen. 2024. Unveiling the tapestry of automated essay scoring: A comprehensive investigation of accuracy, fairness, and generalizability. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Jianhua Zhang and Lawrence Jun Zhang. 2023. Examining the relationship between english as a foreign language learners' cognitive abilities and l2 grit in predicting their writing performance. *Learning and Instruction*, 88.

A GPT prompts used in our experiments

Item	**Description**
Conclusion	Does this text have a concluding section, a summary? Answer with 1 for Yes or 0 for No.
Introduction	Does this text have an introduction? Answer with 1 for Yes or 0 for No.
Main Thesis	Is this text a main thesis, meaning a sentence in a text that takes a clear position? Answer with 1 for Yes or 0 for No.
Position	Does this text discuss all three positions of the task? Either cars that are powered by hydrogen, electricity, or e-fuels, or other task that involves hydroelectric power plants, solar power plants, and wind farms. If all three options are discussed, answer with 1, if not then 0.
Warrant	Do the arguments in the text have an explanation, meaning a more detailed explanation of the argument? If yes answer with 1, if not then 0.

Table 6: GPT prompts

B DARIUS corpus example

Deutsch	**Englisch**
In Norddeutschland wird die Frage gestellt welche klimaneutrale Energiegewinnung gebaut werden soll, um eine Klimaneutralität zu erreichen. Zur Frage kommen Windparks, Solar und Wasserkraftanlagen. Ich finde, dass der Bau von Windparks gefördert werden soll. Mit 45% Wirkungsgrad sind diese schwächer als Wasserkraftanlagen und stärker als Solarparks. Obwohl der Wirkungsgrad mit 45% geringer ist als bei Wasserkraftanlagen, liefert ein Windpark mit 40 GWh pro Jahr mehr Strom als Solarpark und Wasserkraftanlage. Ebenfalls ist der Preis relativ zum Jahresertrag günstig mit 14 Millionen als Solarpark und Wasserkraftanlage. Ebenfalls muss man in Betracht ziehen, dass der Windpark weniger CO2 ausstoßt. Solarpark und Wasserkraftanlage stoßen 35000t und 12000t CO2 und der Windpark nur 8,800t. Jedoch muss man sagen, dass der Windpark nur eine Lebensdauer von 20 Jahren hat. Währenddessen halten Solarparks 30 Jahre und Wasserkraftanlage 80 Jahre. Auf der Ebene der Lokalemissionen besitz der Windpark die meisten Emission mit Hör-, Infraschall und Schattenwurft. Die Wasserkraftanlage wirft keinen Schattenwurf, aber hat trotzdem Hör- und Infraschall. Der Solarpark hat keinen Emissionen jeglicher Art. Zum Schluss komme ich, dass man Windparks fördern sollte, da die Vorteile die Nachteile überwiegen. Sie bieten günstig Strom und verursachen wenig Treibhausgasemissionen, aber man muss anmerken, dass ein Windpark keine hohe Lebensdauer hat, sodass diese öfters erneuert werden müssen, und dass Anwohner und Tiere von diesem belästigt werden können.	In northern Germany, the question is being asked as to which climate-neutral energy generation should be built in order to achieve climate neutrality. The options are wind farms, solar and hydropower plants. I think that the construction of wind farms should be promoted. At 45% efficiency, they are less efficient than hydropower plants and more efficient than solar parks. Although the efficiency of 45% is lower than that of hydropower plants, a wind farm with 40 GWh per year supplies more electricity than solar farms and hydropower plants. The price relative to the annual yield is also lower at 14 million than solar parks and hydroelectric power plants. It must also be taken into account that the wind farm emits less CO2. The solar park and hydropower plant emit 35,000 tons and 12,000 tons of CO2 respectively, while the wind park emits only 8,800 tons. However, it must be said that the wind farm only has a lifespan of 20 years. In contrast, solar parks last 30 years and hydroelectric power plants 80 years. On the level of local emissions, the wind farm has the most emissions with acoustic, infrasound and shadow flicker. The hydropower plant does not cast any shadows, but still has audible and infrasound emissions. The solar park has no emissions of any kind. In conclusion, I believe that wind farms should be promoted because the advantages outweigh the disadvantages. They provide cheap electricity and cause little greenhouse gas emissions, but it should be noted that a wind farm does not have a long lifespan, so they have to be renewed frequently, and that residents and animals can be disturbed by them.

Table 7: Example essay in the DARIUS Corpus, translated via DeepL[4]

Improving Automated Distractor Generation for Math Multiple-choice Questions with Overgenerate-and-rank

Alexander Scarlatos[1*], **Wanyong Feng**[1*], **Digory Smith**[2], **Simon Woodhead**[2], **Andrew Lan**[1]
University of Massachusetts Amherst[1], Eedi[2]
{ajscarlatos, wanyongfeng, andrewlan}@umass.edu
{digory.smith,simon.woodhead}@eedi.co.uk

Abstract

Multiple-choice questions (MCQs) are commonly used across all levels of math education since they can be deployed and graded at a large scale. A critical component of MCQs is the distractors, i.e., incorrect answers crafted to reflect student errors or misconceptions. Automatically generating them in math MCQs, e.g., with large language models, has been challenging. In this work, we propose a novel method to enhance the quality of generated distractors through overgenerate-and-rank, training a ranking model to predict how likely distractors are to be selected by real students. Experimental results on a real-world dataset and human evaluation with math teachers show that our ranking model increases alignment with human-authored distractors, although human-authored ones are still preferred over generated ones.

1 Introduction and Related Work

Multiple-choice questions (MCQs) are commonly used to assess student knowledge across all levels of education, including math, since they can accurately assess student knowledge while being easy to administer and grade at scale (Nitko, 1996; Airasian, 2001; Kubiszyn and Borich, 2016). An MCQ is comprised of a question stem and several answer options. The *question stem* establishes the context and presents a problem for students to solve. Among the options, there exists a *key*, which is the correct answer, and multiple *distractors*, which are the incorrect answers specifically designed to reflect student errors or misconceptions. Although MCQs offer numerous advantages for assessing student knowledge, crafting high-quality distractors poses a significant challenge for teachers and educators. High-quality distractors should be sufficiently challenging so students do not quickly identify them as incorrect answers. Additionally, they should be designed to target specific errors or misconceptions, enticing students who make these errors or hold these misconceptions to choose them. This delicate balance makes the creation of such high-quality distractors a time and labor-intensive endeavor (Kelly et al., 2013).

Earlier works on automatic distractor generation for math MCQs use constraint logic programming (Tomás and Leal, 2013) or manually crafted rules (Prakash et al., 2023) to generate distractors. However, these methods are restricted to template-generated MCQs, which have limited applicability in a broader context. More recent work (Dave et al., 2021) trains a neural network to solve math problems and sample incorrect answers as distractors. Not surprisingly, the generated distractors fail to capture student errors or misconceptions. The most recent works (McNichols et al., 2023a; Feng et al., 2024) explore this task using state-of-the-art large language models (LLMs), such as ChatGPT. The authors experiment with several different approaches, including few-shot in-context learning (Brown et al., 2020) and zero-shot chain-of-thought (CoT) prompting (Wei et al., 2022), showing that LLMs can often generate distractors that are mathematically relevant to the MCQ. However, the overall alignment level with human-authored distractors that are thought to reflect student errors or misconceptions is not high. These works indicate a need to understand what errors or misconceptions are common among students and to use this information to improve the quality of generated distractors.

1.1 Contributions

In this work, we propose a method to enhance the quality of generated distractors through overgenerate-and-rank.[*] Our novel ranking model evaluates the likelihood of each generated distractor being selected by real students. We train the ranking model via direct preference optimization

[*] These authors contributed equally to this work.

[*] Our code is publicly available at https://github.com/umass-ml4ed/distractor-ranking-BEA

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 222–231
June 20, 2024 ©2024 Association for Computational Linguistics

(DPO) on pairwise preference pairs that compare the relative portion of students selecting one distractor over the other. This method can be augmented with existing distractor generation methods.

We validate the effectiveness of this method through extensive experiments on a real-world math MCQ dataset. We find that the ranking model effectively selects distractors that students are more likely to select. In particular, it can improve the generated distractor quality of a fine-tuned `Mistral` model with 7B parameters to a similar level as that of `GPT-4` with CoT prompting, which is rumored to have up to 1T parameters. We also conduct human evaluations where we ask math teachers to rank and rate both LLM-generated and human-authored distractors. Results show that our ranking model's ranking and human ranking correlate with actual ranking defined by the portion of students selecting each distractor to a similar degree. Despite the improvements, LLM-generated distractors still do not match the quality of human-authored ones in reflecting student errors or misconceptions.

2 Methodology

This section contains the details of the task definition and our over-generate-and-rank method.

2.1 Task Definition

We define an MCQ Q as comprising a collection of elements, denoted as $Q = \{s, k, e_k, D, F, P\}$. Specifically, each MCQ includes a question stem s, a key k, an explanation for the key e_k, and a set of distractors D. Each distractor $d_i \in D$ is associated with a feedback message $f_i \in F$ provided to students upon selection. Moreover, for the key and every distractor, we have $p_i \in P$ as the portion of students who select this distractor (among all students who solve the MCQ).[*] Similar to (Qiu et al., 2020), we define the distractor generation task as learning a function g^{dis} that outputs a set of distractors $\hat{D}$ for an MCQ given the question stem, key, and its explanation, i.e., $g^{\text{dis}}(s, k, e_k) \rightarrow \hat{D}$.

2.2 Pairwise Ranking

In order to identify high-quality distractors for overgenerate-and-rank, we propose a ranking function that aligns with how likely distractors are to be selected by students. We define the ranking function as $r(s, k, e_k, d_i) \rightarrow \alpha_i \in \mathbb{R}$, where α_i is a relative score for distractor d_i. Our goal is to train r such that higher scoring distractors are more likely to be selected by students, i.e., $\alpha_i > \alpha_j \rightarrow p_i > p_j$. We achieve this alignment by setting α_i to the log likelihood of d_i under a ranking model $\mathcal{M}$, i.e., $\alpha_i = \log P_{\mathcal{M}}(d_i | s, k, e_k)$, where $\mathcal{M}$ is an autoregressive language model trained to generate distractors that are likely to be selected by students.

We initially fine-tune a model $\mathcal{M}_{\text{SFT}}$, where all distractors in the train set are used as labels for their corresponding questions. While $\mathcal{M}_{\text{SFT}}$ captures the likelihood of a distractor to appear in the data, it does not account for student behavior. We therefore train a model $\mathcal{M}_{\text{DPO}}$ via direct preference optimization (DPO) (Rafailov et al., 2024), using all $\binom{|D|}{2}$ pairs of distractors for each question where the distractor chosen more frequently by students is the preferred one in each pair. This aligns the model with student selections, and is motivated by recent successes of DPO in educational tasks (Scarlatos et al., 2024; Kumar and Lan, 2024).

We validate the effectiveness of this approach by calculating the *ranking accuracy*, i.e., the percentage of distractor pairs in the test set where the predicted ranking agrees with actual student selection percentages. $\mathcal{M}_{\text{SFT}}$ and $\mathcal{M}_{\text{DPO}}$ result in ranking accuracies of 61.60% and 65.84%, respectively; we use the latter in our experiments. While these numbers may appear low (random selection yields 50%), we note that the data is noisy and accuracy improves when there is a higher difference between selection percentages: $\mathcal{M}_{\text{DPO}}$ gets 74.31% accuracy on pairs where the difference between selection percentages is more than 10%. Training details are in Supplemental Material Section B.

2.3 Overgenerate-and-rank and baselines

We instruct a base distractor generation model to overgenerate a set of n distractors, D', such that $n > |D|$. Subsequently, we use our learned ranking model to score each candidate distractor $d_i \in D'$ and choose the $|D|$ distractors with the highest scores as our final set of generated distractors (Kumar et al., 2023). In practice, we use $n = 10$ and have $|D| = 3$ (**Top-3**). We compare our method against two baseline ranking methods: First, we simply randomly select 3 distractors from D' (**Rand-3**). Second, we instruct the base distractor generation model to directly generate exactly 3 distractors (**Only-3**).

[*] All elements within Q, except for P, are formatted as strings, whereas P is formatted as numbers.

3 Experiments

This section provides a comprehensive overview of our dataset, outlines the evaluation metrics and the experimental setup, and details the findings from experiments and human evaluation.

3.1 Dataset

We use a dataset that comprises 1.4K math MCQs sourced from Eedi's content repository[*]. These questions, all written in English, target students aged 10 to 13. Each MCQ includes a question stem, a key with an explanation justifying its correctness, and three distractors, each accompanied by a feedback message clarifying why it is incorrect. Additionally, each option is tagged with the percentage of students choosing that option, computed on an average of 4,000 student responses per question. We split the dataset into training and test sets at an 80:20 ratio. The training set is used to fine-tune the base distractor generation LLM (if necessary) and train the ranking model, while the test set is used for evaluation.

3.2 Evaluation Metrics

We adopt the alignment-based metrics previously introduced in (McNichols et al., 2023a) to assess the degree of alignment between LLM-generated distractors and human-authored ones. There are two binary metrics: **Partial** match, which checks if at least one LLM-generated distractor matches the human-authored ones[*], and **Exact** match, which checks if all LLM-generated distractors match the human-authored ones. There is also one scalar metric: Proportional (**Prop.**) match, which calculates the proportion of LLM-generated distractors that match the human-authored ones. Additionally, to reflect the portion of students selecting each distractor, we introduce a new scalar metric: Weighted Proportional (**W. Prop.**) match (that also has range $[0, 1]$), formally defined as

$$h(D, \hat{D}) = \textstyle\sum_i I(\exists j \text{ s.t. } d_i = \hat{d}_j) \cdot p_i / \sum_i p_i,$$

where I is the indicator function. Intuitively, this metric re-weights each "match" in the Proportional metric such that a match on a distractor that more students select is weighed more heavily than one that less students select. We calculate the values for all metrics by averaging them across all MCQs in the test set and then scaling these values by a factor of 100 to convert them into percentages.

Approach		Partial	Exact	Prop.	W. Prop.
CoT	Top-3	**67.87**	2.53	**32.25**	**36.89**
	Rand-3	47.29	0.00	18.29	19.13
	Only-3	66.43	**3.25**	31.05	35.03
FT	Top-3	**67.15**	1.44	**30.20**	**34.81**
	Rand-3	35.38	0.36	14.20	15.06
	Only-3	60.29	**2.89**	28.28	31.75

Table 1: Results of distractor generation on alignment-based metrics. We see that overgenerate-and-rank (sometimes significantly) improves performance.

3.3 Experimental Setup

Following (McNichols et al., 2023a), we use zero-shot chain-of-thought prompting (**CoT**) with `GPT-4` and fine-tuning (**FT**) with the open-source `Mistral-7B` model as our base distractor generation models. Since our goal is to evaluate the performance of the ranking model, we do not use their in-context learning method, "kNN", because in-context examples leak student selection information into the distractor generation model by showing example distractors that real students frequently select. Consistent with the best practices identified in their work, we represent each target MCQ by concatenating the question stem, the key, and its corresponding explanation. During the distractor generation process, the model must generate a feedback message before the actual distractor. Hyperparameters and model details are listed in the Supplementary Material Section B.

3.4 Results and Discussion

Table 1 shows the performance of both base distractor generation models with different ranking methods across alignment-based metrics. The low Exact match values across methods indicate it is nearly impossible for the LLM to recover the exact 3 human-authored distractors. However, Top3 outperforms both Rand3 and Only3 on all other metrics, which suggests that the trained ranking model is effective at identifying which distractors are more likely selected by students. The gap on the Weighted Proportional metric is bigger than that on the Proportional metric for CoT and FT since the Weighted Proportional metric incorporates student distractor selection percentages, which is what the ranking model trains on. This observation high-

[*] `https://eedi.com/home`

[*] We use the exact string match criterion to align LLM-generated with ground-truth, human-authored distractors.

Comparison	Kendall's Tau
GT Rank vs. Human Rank	0.27
GT Rank vs. Model Rank	0.30
Human Rank vs. Model Rank	0.14

Table 2: Correlation between different rankings on human-authored distractors. Teachers and the ranking model correlate with actual student selection percentages to a similar degree.

lights the advantage of overgenerate-and-rank, suggesting that letting the base distractor generation model to generate a diverse set, casting a wide net, and then using the ranking model to select good ones is an effective approach. Perhaps most importantly, we see that Top3 with FT performs similarly to Only3 with CoT. This observation shows that the ranking model can elevate the performance of a small, open-source LLM (`Mistral-7B`) and make it comparable to a much larger, proprietary LLM (`GPT-4`), which is a promising sign for the potential real-world deployment of automated distractor generation methods in a cost-controlled way.

3.5 Human Evaluation

We conduct human evaluations where we recruit two math teachers with experience teaching grade-school-level math to evaluate distractors. We randomly select 20 MCQs whose Top-3 LLM-generated distractors are completely different from the human-authored ones from the test set. In the first evaluation task, we ask evaluators to rank the quality of human-authored distractors to examine the correlation between teacher judgment (**Human Rank**), the ranking model's ranking (**Model Rank**), and the actual student selection percentages (**GT Rank**). In the second evaluation task, we show evaluators 6 distractors for each MCQ, including 3 LLM-generated distractors and 3 human-authored distractors. We then ask them to rate the overall quality of each distractor to compare LLM-generated distractors (**Top-3 LLM**) with human-authored ones (**Human**), on a 5-point Likert scale, from 1 (least likely to be selected by students) to 5 (most likely). To mitigate potential bias from distractor ordering, the sequence of the distractors was randomized for each MCQ.

Table 2 shows Kendall's Tau correlation (Arndt et al., 1999) between the ground-truth ranking and the human/model ranking. We see that human and model rankings have a weak-to-moderate correlation with the ground-truth ranking. This observation reveals the difficulty of this task since even expert math teachers with years of teaching experience cannot fully anticipate real students' behavior. We also see that human ranking and model ranking have a weak correlation, likely due to humans and LLMs approaching the same problem from different angles; future work can consider a human-AI collaboration approach.

QWK		Average Ratings	
Top-3 LLM	Human	Top-3 LLM	Human
0.66	0.62	2.67 ± 0.96	3.26 ± 1.02*

Table 3: Inter-rater agreement and average ratings on LLM-generated and human-authored distractors. $*$ indicates statistical significance ($p < 0.05$) under a t-test.

Head-to-Head Rating Comparison	Percentage
Top-3 LLM $>$ Human	22%
Top-3 LLM $=$ Human	16%
Top-3 LLM $<$ Human	62%

Table 4: Head-to-head comparison between LLM-generated distractors and human-authored ones. Teachers prefer human-authored ones most of the time.

Table 3 shows the inter-rater agreement among math teachers, measured in quadratic weighted Kappa (QWK) (Brenner and Kliebsch, 1996), and their average ratings for both LLM-generated distractors and human-authored ones. We see that human-authored distractors are preferred with statistical significance, and the inter-rater agreement is moderate-to-substantial. However, we note that since the 20 selected MCQs in our evaluation are the ones where none of the top-3 LLM-generated distractors match human-authored ones, this result may downplay the effectiveness of LLMs because they must generate plausible distractors that are not already included in the human-authored ones.

We additionally compare the LLM-generated and human-authored distractors head-to-head, using average distractor rating across evaluators between each LLM-generated distractor and each human-authored distractor for each question (resulting in 9 comparisons per question). Table 4 shows the percentage of cases where LLM-generated distractors win, lose, or tie to human-authored ones. We see that even though human-authored distractors are preferred the majority of the time, there is a sizeable portion of LLM-generated distractors that are equal to or preferred over human-authored distractors. This result implies that LLMs can generate some high-quality distractors that can be used to enhance the quality of human-authored ones.

4 Conclusions and Future Work

In this paper, we propose an overgenerate-and-rank method for generating distractors for math MCQs. We train a ranking model to predict which distractors students would select more often, and this ranking model can be applied to any existing distractor generation method. We experimentally validate its performance on a real-world dataset and test its limitations through human evaluation.

Avenues for future work include but are not limited to further improving the ranking model through a student-specific distractor selection prediction objective that considers their knowledge state (Liu et al., 2022), developing a human-in-the-loop approach for distractor selection percentage prediction, and using the same approach for feedback generation (Scarlatos et al., 2024). Finally, extending our work from multiple-choice questions to open-ended questions is important, since open-ended student responses contain much more detailed information on their errors (Zhang et al., 2021, 2022; McNichols et al., 2023b).

Limitations

First, due to limited resources, we only performed human evaluation on the human-authored distractors and the Top-3 LLM-generated distractors. However, this does not allow us to determine if our overgenerate-and-rank approach is better than generation baselines from a human evaluation perspective. We also acknowledge that our human evaluation sample size is small, and should ideally be increased for future studies. Second, while we have evidence that our method enhances the quality of LLM-generated distractors, a notable difference remains between the quality of LLM-generated distractors and human-authored ones. To make LLM-generated distractors viable for deployment in real educational settings, it is necessary to further investigate how to improve their overall quality. Third, our first human evaluation result shows that even experienced math teachers cannot anticipate real student behavior accurately. A more precise evaluation for LLM-generated distractors would involve deploying them in actual tests and observing student behavior. However, this process can be significantly complicated and time-consuming, and should only be performed when there is reasonable evidence that generated distractors might be of similar quality to human-authored ones.

Ethical Considerations

Our work uses the overgenerate-and-rank method to improve the quality of LLM-generated distractors. We believe that our work could potentially reduce the time educators and teachers spend creating math MCQs, enabling them to focus more on teaching and engaging with students. However, we acknowledge that potential biases within LLMs may exist, which could cause the LLM-generated distractors to contain incorrect or potentially harmful information. Therefore, we strongly recommend that educators and teachers review the quality of LLM-generated distractors thoroughly before deploying them in actual tests for students.

References

Peter Airasian. 2001. *Classroom assessment: Concepts and applications.* McGraw-Hill, Ohio, USA.

Stephan Arndt, Carolyn Turvey, and Nancy C Andreasen. 1999. Correlating and predicting psychiatric symptom ratings: Spearmans r versus kendalls tau correlation. *Journal of psychiatric research*, 33(2):97–104.

Hermann Brenner and Ulrike Kliebsch. 1996. Dependence of weighted kappa coefficients on the number of categories. *Epidemiology*, pages 199–202.

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. abs/2005.14165.

Neisarg Dave, Riley Bakes, Barton Pursel, and C Lee Giles. 2021. Math multiple choice question solving and distractor generation with attentional gru networks. *International Educational Data Mining Society.*

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, 35:30318–30332.

Wanyong Feng, Jaewook Lee, Hunter McNichols, Alexander Scarlatos, Digory Smith, Simon Woodhead, Nancy Otero Ornelas, and Andrew Lan. 2024. Exploring automated distractor generation for math multiple-choice questions via large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024, arXiv preprint arXiv:2404.02124*.

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*.

Kim Kelly, Neil Heffernan, Sidney D'Mello, Namais Jeffrey, and Amber C. Strain. 2013. Adding teacher-created motivational video to an its. In *Proceedings of 26th Florida Artificial Intelligence Research Society Conference*, pages 503–508.

Tom Kubiszyn and Gary Borich. 2016. *Educational testing and measurement.* John Wiley & Sons, New Jersey, USA.

Nischal Ashok Kumar, Nigel Fernandez, Zichao Wang, and Andrew Lan. 2023. Improving reading comprehension question generation with data augmentation and overgenerate-and-rank. *arXiv preprint arXiv:2306.08847*.

Nischal Ashok Kumar and Andrew Lan. 2024. Improving socratic question generation using data augmentation and preference optimization.

Naiming Liu, Zichao Wang, Richard Baraniuk, and Andrew Lan. 2022. Open-ended knowledge tracing for computer science education. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3849–3862.

Hunter McNichols, Wanyong Feng, Jaewook Lee, Alexander Scarlatos, Digory Smith, Simon Woodhead, and Andrew Lan. 2023a. Automated distractor and feedback generation for math multiple-choice questions via in-context learning. In *NeurIPS'23 Workshop on Generative AI for Education (GAIED)*.

Hunter McNichols, Mengxue Zhang, and Andrew Lan. 2023b. Algebra error classification with large language models. In *International Conference on Artificial Intelligence in Education*, pages 365–376.

Anthony J. Nitko. 1996. *Educational assessment of students.* Prentice-Hall, Iowa, USA.

Vijay Prakash, Kartikay Agrawal, and Syaamantak Das. 2023. Q-genius: A gpt based modified mcq generator for identifying learner deficiency. In *International Conference on Artificial Intelligence in Education*, pages 632–638. Springer.

Zhaopeng Qiu, Xian Wu, and Wei Fan. 2020. Automatic distractor generation for multiple choice questions in standard tests. *arXiv preprint arXiv:2011.13100*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Alexander Scarlatos, Digory Smith, Simon Woodhead, and Andrew Lan. 2024. Improving the validity of automatically generated feedback via reinforcement learning. In *International Conference on Artificial Intelligence in Education, arXiv preprint arXiv:2403.01304*.

Ana Paula Tomás and José Paulo Leal. 2013. Automatic generation and delivery of multiple-choice math quizzes. In *Principles and Practice of Constraint Programming: 19th International Conference, CP 2013, Uppsala, Sweden, September 16-20, 2013. Proceedings 19*, pages 848–863. Springer.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

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. abs/1910.03771.

Mengxue Zhang, Sami Baral, Neil Heffernan, and Andrew Lan. 2022. Automatic short math answer grading via in-context meta-learning. *International Educational Data Mining Society*.

Mengxue Zhang, Zichao Wang, Richard Baraniuk, and Andrew Lan. 2021. Math operation embeddings for open-ended solution analysis and feedback. *International Educational Data Mining Society*.

Supplementary Material

A Distractor Generation Examples

Question Stem		
fifty five thousand subtract twenty three thousand equals		
Key		
32,000		
Human-authored Distractors		
22,000	23,000	3,200
Only-3 LLM-generated Distractors		
52,000	-32,000	78,000
Top-3 LLM-generated Distractors		
33,000	-32,000	30,000

Table 5: Representative question with distractors from humans, GPT-4 generating only 3, and GPT-4 after selecting the top 3 with our ranking model.

B Experimental Details

We take several measures to ensure that generated distractors are distinct and different from the key. For CoT, we prompt GPT-4 to generate 15 distractors and eliminate duplicates and those identical to the key. For the rest of MCQs lacking 10 distinct distractors, we prompt GPT-4 again to generate 15 new distractors, instructing it to avoid producing previously generated distractors by including them in the prompt. We supplement the existing distractors with the newly generated distractors, ensuring the total number of distinct distractors reaches 10. For the MCQs that still lack 10 distinct distractors (which are few), we add the word "placeholder" as distractors. We use greedy decoding for all previous steps. When overgenerating distractors with our fine-tuned model, we generate 3 distractors 5 times using nucleus sampling for each MCQ, setting temperature $= 1$ and top_p $= 0.9$. If we do not get 10 unique distractors, we generate 5 more times with top_p $= 1.0$ to ensure greater diversity. When generating only 3 distractors, we use beam search with num_beams $= 5$. If we do not get 3 unique distractors, we then generate with nucleus sampling twice with top_p $= 0.9$ and take the first 3 unique distractors.

For the fine-tuned distractor generation model and the pairwise ranking model, we use the `mistralai/Mistral-7B-v0.1` model from HuggingFace (Wolf et al., 2019) and load the model with 8-bit quantization (Dettmers et al., 2022). We train LoRA adapters (Hu et al., 2021) on the `q_proj`, `v_proj`, `k_proj`, and `o_proj` matrices, setting $r = 32$, $\alpha = 16$, dropout $= 0.05$. We train using the AdamW optimizer with a virtual batch size of 64 using gradient accumulation and do early stopping on a random 20% subset of the train set. For the distractor generation model we use a learning rate of 5e-5 and train for 15 epochs, and for the pairwise ranking model we use a learning rate of 3e-5 and train for 5 epochs. For DPO training on the pairwise ranking model, we set $\beta = 0.5$ and use $\mathcal{M}_{\text{SFT}}$ as the reference model. We train all models on a single NVIDIA RTX A6000 GPU.

C Human Evaluation Details

In this work, we obtained approval from the ethics review board for human evaluation. We show the evaluation instructions to human evaluators in Table 9. We do not provide any compensation for human evaluators because their participation is entirely voluntary and we appreciate their contribution to this work.

D Prompt Format

We provide the prompts for CoT, FT, and pairwise ranking model below. We use <> to indicate that a variable is filled in dynamically.

Prompt	You are provided with a math question, correct answer, and the explanation of correct answer. Your task is to use the following template to create 15 unique incorrect answers (distractors) to be used as multiple-choice options for a middle school math multiple-choice question. Before generating each distractor, include a concise explanation to clarify for students why that is not the correct answer. Make sure each distractor is clearly different from the correct answer and distinct from each other, this is very important! [Template] Distractor1 Feedback: Distractor1: Distractor2 Feedback: Distractor2: Distractor3 Feedback: Distractor3: Distractor4 Feedback: Distractor4: Distractor5 Feedback: Distractor5: Distractor6 Feedback: Distractor6: Distractor7 Feedback: Distractor7: Distractor8 Feedback: Distractor8: Distractor9 Feedback: Distractor9: Distractor10 Feedback: Distractor10: Distractor11 Feedback: Distractor11: Distractor12 Feedback: Distractor12: Distractor13 Feedback: Distractor13: Distractor14 Feedback: Distractor14: Distractor15 Feedback: Distractor15: Question: <question> Explanation: <explanation> Answer: <answer>

Table 6: Prompt for chain-of-thought distractor generation with GPT-4.

Prompt	You are provided with a math question, correct answer, and the explanation of correct answer. Your task is to generate 3 unique incorrect answers (distractors) to be used as multiple-choice options for a middle school math multiple-choice question. Before generating each distractor, include a concise explanation for students to clarify why that is not the correct answer. Ensure each distractor is different from the correct answer and distinct from the others; this is very important! Question: <question> Explanation: <explanation> Answer: <answer>

Table 7: Prompt for fine-tuning with Mistral.

Prompt	A teacher assigns the following math question to a class of middle school students. Question: <question> Solution: <solution> Correct answer: <correct answer> Generate a distractor for this question that targets some student misconception. Distractor: <distractor>

Table 8: Prompt for pairwise ranking model.

E Human Evaluation Instructions

You are provided with two tasks

The first task (rank) consists of 20 items, each containing a question stem and three distractors. For each item, you are asked to rank the three distractors based on the assessment of how often they will be selected by real students, from most frequent to least frequent. The items for this task can be accessed in the rank.csv file.

Example:
Question: How do you write 4.6 as a percentage?
Distractor 1 (id = 1): 46%
Distractor 2 (id = 2): 0.046%
Distractor 3 (id = 3): 4.6%
Best distractor id: 1
Second best distractor id: 3
Third best distractor id: 2

The second task (rate) also consists of 20 items, each containing a question stem and six distractors. For each item, you are asked to rate the likelihood of each distractor being selected by students on a 5-point scale independently: 5 - most likely, 4 - likely, 3 - average, 2 - not likely, and 1 - least likely. The items for this task can be accessed in the rate.csv file

Example:
Question: How do you write 4.6 as a percentage?
Distractor: 46%
Rating: 4

Table 9: Instructions given to human evaluators for evaluating distractors.

Identifying Fairness Issues in Automatically Generated Testing Content

Kevin Stowe[1], **Benny Longwill**[1], **Alyssa Francis**[1]
Tatsuya Aoyama[2][*], **Debanjan Ghosh**[1], **Swapna Somasundaran**[1]
[1]Educational Testing Service (ETS), Princeton, New Jersey
[2]Georgetown University

Abstract

Natural language generation tools are powerful and effective for generating content. However, language models are known to display bias and fairness issues, making them impractical to deploy for many use cases. We here focus on how fairness issues impact automatically generated test content, which can have stringent requirements to ensure the test measures only what it was intended to measure. Specifically, we review test content generated for a large-scale standardized English proficiency test with the goal of identifying content that only pertains to a certain subset of the test population as well as content that has the potential to be upsetting or distracting to some test takers. Issues like these could inadvertently impact a test taker's score and thus should be avoided. This kind of content does not reflect the more commonly-acknowledged biases, making it challenging even for modern models that contain safeguards. We build a dataset of 601 generated texts annotated for fairness and explore a variety of methods for classification: fine-tuning, topic-based classification, and prompting, including few-shot and self-correcting prompts. We find that combining prompt self-correction and few-shot learning performs best, yielding an F1 score of 0.79 on our held-out test set, while much smaller BERT- and topic-based models have competitive performance on out-of-domain data.[1]

1 Introduction

Large language models (LLMs) have become ubiquitous in the space of natural language generation (NLG) due to recent advances in model capability (Minaee et al., 2024). However, these improvements come with the potential for various negative societal impacts. These negative impacts include the generation of misinformation/propaganda, allocation harms of systems providing benefits only to certain groups of people, and representational harms revolving around bias and stereotyping. Natural language processing (NLP) models–including LLMs–are known to reflect and repeat harmful biases and stereotypes (Hosseini et al., 2023; Bender et al., 2021; Hovy and Prabhumoye, 2021; Nadeem et al., 2021), and research into how the community addresses the societal harms engendered by NLP technology is critical (Wang et al., 2024; Dev et al., 2022; Blodgett et al., 2020).

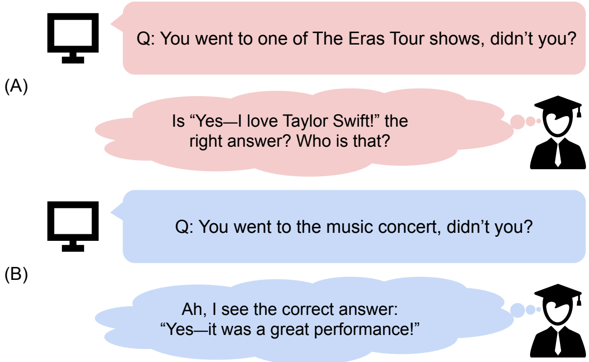

Q: You went to one of The Eras Tour shows, didn't you?
(A)
Is "Yes—I love Taylor Swift!" the right answer? Who is that?
Q: You went to the music concert, didn't you?
(B)
Ah, I see the correct answer: "Yes—it was a great performance!"

Figure 1: In (A), the generated question requires knowledge of what The Eras Tour is to identify the correct answer. Even native English speakers would likely not be able to identify the correct response if they were not familiar with Taylor Swift. In (B), the generated question does not require specific background knowledge, so test takers would not need to use specialized knowledge to identify the correct answer. Our goal is to identify and filter content like (A) to help ensure fair testing.

Many of these types of bias in language generation are well-studied. Biases based on gender (Nemani et al., 2024; Devinney et al., 2022; Strengers et al., 2020; Wan et al., 2023), race (Das and Balke, 2022; Field et al., 2021), nationality (Venkit et al., 2023), and disability (Venkit et al., 2022) have been identified in language models, and many modern LLMs incorporate deliberate safeguarding measures in an attempt to alleviate these

[*] Work done while at ETS

[1] Code and dataset available at `https://github.com/EducationalTestingService/item-generation-fairness`.

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 232–250
June 20, 2024 ©2024 Association for Computational Linguistics

issues (OpenAI et al., 2023; Anil et al., 2023).

In the area of language assessment, there exists a tangential set of issues regarding fairness to test takers and score users (Educational Testing Service, 2022). These issues are particularly dangerous when applied to language learning and assessment; tests with inherent biases have the potential to compromise the validity of the test. Therefore, content that is irrelevant to the skills and abilities the test is intended to measure should be avoided (Figure 1). This includes content that could disadvantage anyone based on their culture, location, or experiences (e.g., focusing on barbeques on the 4th of July could disadvantage test-takers who are unfamiliar with U.S. culture); their emotions (e.g., health hazards and diseases can evoke negative emotional responses among some people); their worldviews (e.g., luxury cruises or designer clothing may make some people feel excluded); and other factors. We refer to these types of issues as **fairness** issues. Knowing how to better understand, detect, and mitigate bias related to fairness in NLG not only raises awareness of the issue but also enables researchers and developers to create more fair and inclusive NLP systems, evaluation metrics, and datasets in the language assessment space.

Our goal is to build a system for identifying fairness-violating content in automatically generated texts. It is of course still necessary to have human review and revision of the content, but by adding a filtering process after generation and before manual review, we can significantly reduce the time taken for reviewing and the chance that fairness-related content is mistakenly allowed. To accomplish this goal, we explore four different approaches: fine-tuning, topic-based classification, few-shot prompting, and prompt-self correction.

Our methods need to adapt to new contexts: our definition of fairness is operationally defined by the particular testing context, and may not apply to others, so the guidelines, prompts, and models may not apply generally to new contexts. For this reason, we assess our methods on two held-out test sets and analyze how our methods could be applied to new contexts. We release our resulting dataset, consisting of 620 samples, of which 19.4% contain fairness issues[2], to facilitate improvements in the fairness-detection community.

Our contribution consists of the following:

1. We define a new fairness problem around issues faced in developing fair testing content.
2. We release a dataset of 601 samples for use in evaluating fairness detection methods.
3. We analyze the relative effectiveness of a variety of well-known classification techniques.
4. We provide a new mechanism for prompting self-correction, which yields significant improvements over other prompting strategies.

We start with data collection and analysis. We collect 620 samples over seven different types of content generated using LLM prompting. We annotate each sample and assess whether it contains a fairness issue, and if it does, whether that fairness issue pertains to *knowledge, skill, or expertise* or *emotion* (more on these categories and how they relate to fairness in Section 3). We then use this dataset to experiment with a series of models for classifying fairness issues.

We show that fine-tuning and filtering by topic can be cheap and effective options, although prompting strategies with GPT4 tend to be more effective. Few-shot prompting along with self-correcting prompt strategies yield strong performance with relatively little data, and combining both yields the best results on our in-domain test set, with an F1 score of .773. Interestingly, using a shorter, more generic prompt combined with our self-correction method yields the best result on our out-of-domain test set, with an F1 score of .462.

2 Related Work

Bias, fairness, and responsible AI has been at the forefront of education technology, with contemporary research focusing on automated scoring, writing assistance, and other nuances of applying NLP technology to this sensitive domain (Mayfield et al., 2019; Loukina et al., 2019). Baffour et al. (2023) find that assisted writing tools may exhibit moderate bias depending on the task, while Wambsganss et al. (2023) found no significant gender bias difference in writing done with and without automated assistance. Wambsganss et al. (2022) explore bias in educational tools for German peer review, and Kwako et al. (2023, 2022) propose novel methods for detecting bias in automated scoring algorithms.

We are specifically interested in applications to language generation, and there is also substantial

[2] Each sample we used was rejected for deployment in actual tests. Using rejected samples for our experiments allows us to release the dataset: accepted stimuli cannot be made public.

work in using LLMs and other NLP technology to generate content for educational assessments (Laverghetta Jr. and Licato, 2023; Gonzalez et al., 2023; Heck and Meurers, 2023; Uto et al., 2023; Tack et al., 2023; Stowe et al., 2022). However, this work largely fails to address bias and fairness issues in content generation. Our work is specifically focused on fairness issues in automatically generated language testing content.

In the context of language models, fairness and bias have emerged as critical concerns. Existing detection and mitigation tools generally diverge from our work: some are overly domain-specific like the focus on news articles in Raza et al. (2024), while others are focused on assessing issues within the language models and datasets (Bellamy et al., 2018), rather than the outputs. Other works rely on retrospective metrics that assess a model's fairness through aggregated predictions and subgroup analysis, and/or focus on classification rather than generation problems (Weerts et al., 2023; Wiśniewski and Biecek, 2022; Saleiro et al., 2019). Although these tools enhance transparency and accountability for evaluating language model issues, they fundamentally differ from our bias detection approach tailored for evaluating generated text in real-time for a production environment.

3 Problem Motivation

In the language testing context, we face a unique set of fairness challenges in generating content. Specifically, fair testing requires content that does not contain irrelevant factors that negatively impact the assessment of a test taker.

A primary concern is to ensure that the test content measures only what it is intended to measure. For English-language proficiency tests, this means that the test must measure only the skills and abilities needed to communicate effectively in English, and not other constructs such as background knowledge of specific jobs, events, or cultures.

Consider the following question and an example of a response to that question:

- Question: You went to one of The Eras Tour shows, didn't you?
- Response: Yes–I love Taylor Swift!

If the task were to identify whether the response is an appropriate response to the question, even some native English speakers would likely get it wrong. This is because, in addition to needing to know features of English proficiency (in this case, the ability to infer gist, purpose, and basic context based on information stated in short spoken texts), one would also need to know about Taylor Swift and her concert tour. Thus, those familiar with Taylor Swift would have an unfair advantage in identifying the correct answer.

Eliminating the fairness issue for this type of question would result in the following revision:

- Question: You went to the music concert, didn't you?
- Response: Yes–it was a great performance!

In addition to avoiding testing outside knowledge, it is also important that language proficiency tests do not include content that is offensive or disturbing. For example, the following question and response refer to serious health issues, which have the potential to evoke deep negative emotions.

- Question: Did you hear that Luis has been hospitalized?
- Response: No, but I knew he had a bad case of Covid-19.

Content like this that could prompt strong feelings of anger, sadness, or anxiety should be avoided because it could derail a test taker's concentration, resulting in lower performance on the test. How a test taker interacts with this test content may tell more about their ability to concentrate under emotional strain than about their ability to identify a response's linguistic appropriateness. Eliminating this construct-irrelevant content helps to ensure that the test measures only the skills and abilities it is intended to measure.

4 Methods

Our goal is to detect whether a generated stimulus contains an issue as a binary classification task. We build a dataset of texts labeled for potential fairness issues and explore potential detection methods.

4.1 Dataset

Our goal is to identify and mitigate these fairness issues in testing content. We build a dataset spanning seven different item or task types from standardized English language proficiency tests all generated using GPT4 (OpenAI et al., 2023). Item and task types can contain up to four components: the stimulus (main text the question is based on), stem

Item/Task Type	Total	Fairness	KSA	Emotion
Read a Text Aloud	304	55	24	39
Talks	91	12	6	6
Text completion	84	26	11	19
Respond to Questions Using Information Provided	56	10	5	5
*Conversations	41	8	5	4
*Respond to a Written Request	25	7	6	1
Total	601	118	57	74

Table 1: Item/task types and annotations for fairness issues. Each has a binary annotation (fairness issue/no fairness issue) and is tagged as containing a KSA issue or an Emotion issue. Types marked with '*' are held out for testing as an "out-of-domain" dataset, and not used for any training/evaluation.

(question asked about the stimulus), key (the correct answer to the stem), and distractors (a set of alternative answers that are incorrect).

Fairness issues are possible in all components, but we focus on only the stimuli, which are typically the longest, most feature-rich components of the test content, and thus are most likely to reflect fairness and bias issues. Issues in the stimuli can leak through to other components, making the stimulus the source of the majority of fairness issues.

Annotation For each stimulus, we aim to identify whether or not the stimulus contains fairness/bias issues, and if so, what type of issue is present. We start with a dataset of automatically generated stimuli. These stimuli were generated using prompting and different versions of GPT: the prompts were iteratively improved with the goal of improving the overall quality of the stimuli. During this process, each stimulus was evaluated by the test's content development experts. For this work, the stimuli used were rejected by the reviewers, allowing us to provide them publicly and explore their use for fairness detection. These rejected stimuli typically have the relevant language and structure, so our goal is to identify which of those stimuli were rejected (at least in part) for fairness reasons. We employ content development experts to annotate these samples, yielding a binary classification between non-fairness and fairness-related rejections.

However, there are different ways for bias and fairness considerations to impact individual stimuli. To better understand and mitigate these issues, we separated them into two main categories:

- *Knowledge, Skill, and Ability (KSA)*: content that contains construct-irrelevant information that may be unavailable to test takers in different environments or with different experiences and abilities. These include content with reference to specific skills, regionalisms, or unfamiliar contexts.
- *Emotion*: content in which language, scenarios, or images are likely to cause strong emotions that may interfere with the ability of some groups of test takers to respond. These include offensive, controversial, upsetting, or overly negative content.

Each sample that is flagged for fairness is annotated for one or both of these categories. This allows further analysis to address these specific fairness categories and to better understand the impact of specific fairness issues.

Our dataset is comprised of stimuli from seven different item and task types: a summary of the collected data is shown in Table 1, with examples for each type in Appendix A. These stimuli represent various structures, depending on the item/task type: Read a Text Aloud, Talks, and Text Completion stimuli are short text paragraphs, while Conversation stimuli involve turns between two or more speakers. Respond to Questions Using Information Provided and Respond to a Written Request task stimuli are structured content: the generation process creates text that is filled into a structured template; we use only the raw text.

Overall we collect 601 samples, of which 19.6% exhibit evidence of fairness issues, with 9.5% reflecting KSA issues and 12.3% Emotion issues. We build a validation set of 48 samples reflecting a balance of the item and task types from the training types (Read a Text Aloud, Talks, Text Completion, and Respond to Questions Using Information Provided), and an equal-sized "in-domain" dataset from these stimuli is held separately for testing. These datasets contain an even number of positive and negative classes for fairness evaluations. As our goal is to be able to identify positive cases where fairness issues exist, we intend for our validation and test sets to have a substantial number of this class. We use the two remaining types (Conversations, Respond to a Written Request) as a separate "out-of-domain" test set to evaluate performance on unseen content.

4.2 Experiments

We experiment with standard transformer-based classification baselines, topic detection, and a variety of GPT4-based prompting, including methods for automatic prompt-self correction. We describe each method below: each is tuned on the validation set, and we report the best model performance on that set. We then evaluate model performance on two separate test sets in Section 5.

Classification with Fine-Tuning We fine-tune standard pre-trained transformer models for sequence classification. We experiment with `bert-base-cased`, `bert-large-cased` (Devlin et al., 2019), `roberta-base`, (Liu et al., 2019) and `deberta-base` (He et al., 2021) models. We perform a hyperparameter search on our validation set for each model, finding that a learning rate of `2e-5` over 2-4 epochs generally performs best, and report results using the model with the best performance.

Topic-Based Filtering We observe that many samples are flagged for fairness due to the topic of the material: many topics contain content that violates our fairness guidelines directly, while others are simply more likely to include unacceptable content. Motivated by this, we explore topic detection as a method for identifying fairness issues.

We first identify topics found within the data. We use the topic modeling framework BERTopic (Grootendorst, 2022) to extract topic representations from two sources of training data: (1) all samples from the training partition of our dataset and (2) our fairness guidelines. In this method, SentBERT (Reimers and Gurevych, 2019) converts each training document into a dense vector representation which are then grouped by semantic similarity, creating clusters that represent different topics. For each of the two training sets, topic descriptions made up of the most important words in a cluster are generated for the clusters containing at least five supporting documents. We manually assess each topic description for themes that should be avoided based on their relation to known fairness issues and which topics are acceptable. Finally, for each unseen sample in test and validation datasets, we make predictions based on the single nearest topic cluster. If a sample falls within the boundaries of restricted topics, it is classified as a violation.

Results for these methods are shown in Table 2. The fine-tuned bert-based models perform fairly well, with F1 scores for `bert-large-cased` and `roberta-base` both around 0.65, and `deberta-base` showing exceptional performance with an F1 score of 0.77. The Topic-Based Filtering models are worse, with the data-based system yielding an F1 score of 0.58. In all cases, precision is much higher than recall; these models are conservative with predictions.

Fine-tuning

Model	Prec	Rec	F1
`bert-base-cased`	1.00	0.29	0.45
`bert-large-cased`	0.92	0.50	0.65
`roberta-base`	0.92	0.50	0.65
`deberta-base`	1.00	0.63	0.77

Topic-based Filtering

Model	Prec	Rec	F1
Topic-data	0.79	0.46	0.58
Topic-guidelines	1.00	0.04	0.10

Table 2: Results for fine-tuning (above) and topic detection (below) on the validation set.

4.3 Prompting

We initially experiment with five different "base" prompts. We pair these with stimuli and use GPT4 to return "True" if the stimulus contains a fairness issue and "False" otherwise. These prompts represent different strategies[3]:

- **GENERIC (SHORT) 53 tokens**: Drawing from general knowledge of fairness and bias in LLMs, we write a generic prompt designed to combat attested LLM biases. This prompt is designed as a weak baseline. Our goal is to determine if a short, simple prompt can capture relevant issues, and whether or not it can be easily improved via self-correction or few-shot learning (Sections 4.3 and 4.3)
- **GENERIC (LONG) 191 tokens**: This is a longer, more detailed version of the above, containing nearly 200 tokens.
- **GUIDELINE (SHORT) 197 tokens**: We craft a prompt based on guidelines for writing fair assessments. Using documentation that defines what constitutes fair assessment items and how to write them, we build a prompt capturing the important components of a fair question. The goal of this prompt is to determine whether human-written guidelines based on theoretical issues will accurately capture these issues in real data.

[3] Prompts in Appendix B.

- GUIDELINE (LONG) **1081 tokens**: We construct a "long" version of the previous guidelines by summarizing the entire fairness guidelines with the help of GPT4, asking for concise versions of relevant sections and combining them into a document that fully captures all the relevant aspects of the guidelines. This prompt is our longest, but still fully based on documentation. The goal of this prompt is to determine the efficacy of a longer, more comprehensive prompt.
- DATA-DRIVEN **142 tokens**: We craft a prompt based on annotations in our data. We identify which topics and language cause fairness issues and build the prompt to reflect how they might generalize to unseen item/task types and topics. This method is hypothesized to be the most effective, as it will address known issues in the data but may not extend to unseen data, as it is built specifically around the given training samples.

These prompts are run through GPT4 via the Azure interface (OpenAI et al., 2023). Each prompt was updated manually to correct obvious potential issues. Our goal here is not to overoptimize prompt writing, which could lead to overfitting the validation set, but rather to develop a generic prompt likely to be effective for both known fairness issues and novel issues possible in generated content.

Initial experiments on the validation set revealed two insights: the GENERIC (LONG) prompt performs similarly to the GENERIC (SHORT) in all cases, and the GUIDELINE (LONG) prompt is ineffective. We therefore focus our efforts on the three other prompts: GENERIC (SHORT) GUIDELINE (SHORT) and DATA-DRIVEN.

A primary issue in identifying the fairness issues we are concerned with is that they are only applicable in the context of certain testing environments. The guidelines and data we use are specific to a certain test; for other tests, other environments, and other contexts, different definitions of fairness and different alignments will be applicable. For this reason, the ability to adapt to new environments is critical in applying fairness constraints. We aim to build a system that can learn to adapt to new guidelines with minimal information.

We explore several methods to improve the robustness of prompting: few-shot prompting and self-correcting prompting techniques.

Few-Shot Prompting We follow few-shot prompting strategies that have been successful for LLM classification tasks (Logan IV et al., 2022; Brown et al., 2020). For each of our base prompts, we provide a set number n of additional samples for both the positive and negative classes. Each of the samples contains the text of the stimulus along with a brief description of why it does or does not violate our fairness guidelines.

Prompt Self-Correction We leverage our annotated data using a prompt self-correction strategy (Figure 2). We start with a base prompt and iterate through a batch of samples from our training data. For each sample, we insert text into the prompt, perform a pass through GPT4, and evaluate the resulting output. If it is correct, we continue. If not, we instruct the LLM to make adjustments to the prompt, either to disallow the sample (when the error is a false negative) or to allow it (when the error is a false positive), using the following prompts:

- *Positive sample incorrectly classified as negative*: The previous text was classified as not having any fairness issues, returning False. However, it did have an issue. Think about what topics or issues in the text might be unfair. Improve the original fairness instructions by adding or modifying restrictions so to account for this text.
- *Negative sample incorrectly classified as positive*: The previous text was classified as having a fairness issue, returning True. However, this item has no issue. Think about why this text was classified as unfair. Improve the original fairness instruction to allow this item by removing or revising restrictions.

This process is run up to e epochs, or stopped early if accuracy reaches 1 or the predictions are stable. We run over b batches of n samples randomly drawn from the training data, using the best-scoring prompt from the final batch for evaluation.[4]

Combining Few-Shot and Self-Correction Few-shot and self-correction are inherently complimentary, as the self-correction method returns an optimized prompt and few-shot learning reinforces it by providing in-domain examples. We combine them by concatenating additional few-shot learning samples to the self-correcting prompts.

[4] For an example of the process, see Appendix C.

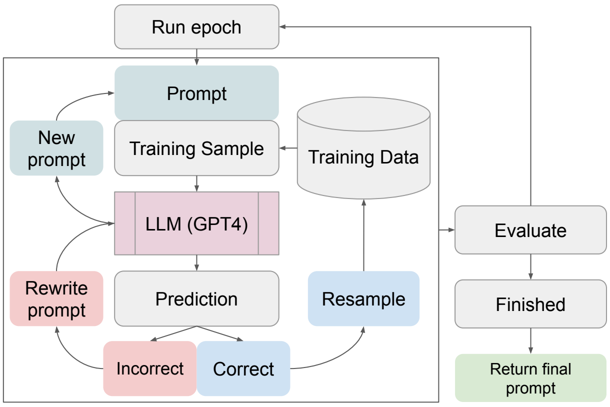


Figure 2: Self-correcting prompt strategy. Data is run through the prompt. If the result is correct, we continue; otherwise, we instruct the LLM to correct the prompt.

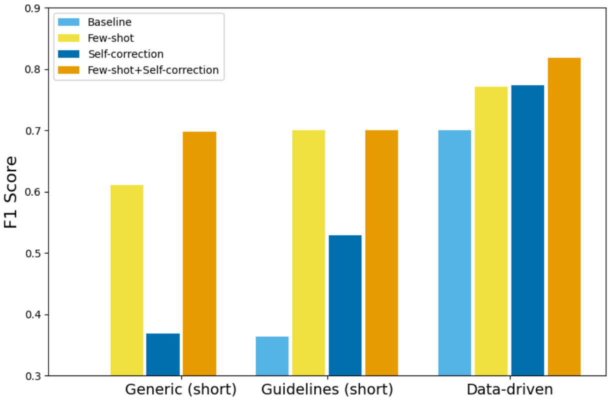


Figure 3: F1 scores on the validation set for each prompting method. Note that for GENERIC (SHORT) the F1 score was 0. Full results in Appendix D.

For each of these improvements to prompting, we perform a hyperparameter search over the number of total training/few-shot samples and batch size. We experiment with the GENERIC (SHORT) GUIDELINE (SHORT) and DATA-DRIVEN prompts.[5] We hypothesize the GENERIC (SHORT) and GUIDELINE (SHORT) prompts should be able to benefit quickly from adaptive methods, while the DATA-DRIVEN prompt should be nearly optimized, as it is already based on observations from the data.

We use the validation set to tune the prompts and parameters to optimize the F1 score for each method. Note that for all prompting strategies, the temperature is set to zero; the prompts should only return True or False. Figure 3 shows the best results on the validation set. We explore each model's effectiveness on unseen data in Section 5.

The base generic prompt fails, as the traditional bias and stereotyping issues are less likely to occur in our generated content, and the fairness issues we are concerned with are unlikely to be deemed as problematic out of context. Using a simplified version of our guidelines yields a 0.36 F1 score for identifying fairness issues. The DATA-DRIVEN based on observations in the training data yields much better results (0.70 F1). However, this may not extend well to novel cases, as the prompt is driven purely by our validation data.

Few-shot learning displays some interesting properties: we see significant improvements across all three prompts, using three samples. (This yielded the best results across all validation runs). Even the minimal GENERIC (SHORT) prompt rises to over 0.60 F1 with minimal few-shot prompting.

We see small improvements over the baseline using prompt self-correction for all three prompts. For the DATA-DRIVEN prompt, results using self-correction equal those using few-shot learning. This aligns with previous work showing that language models themselves tend to write better prompts (Fernando et al., 2023): after only a few iterations of self-correction, the DATA-DRIVEN prompt surpasses the performance of a human-written prompt, even in cases where the human describes the dataset explicitly.

Combining self-correction and few-shot learning yields improvements over base prompts and few-shot prompting alone. This approach yields the best results for all three prompts, with the best-performing model being the DATA-DRIVEN prompt with self-correction and few-shot learning. This may be due to overfitting, however: the prompt is written to reflect the data. To explore the efficacy of these methods on unseen data, we evaluate them on our two held-out test sets.

5 Test Results

The previous experiments describe our attempts to identify the best-performing model for fairness classification on our validation set. Our goal is to develop a system that generalizes. For this, we evaluate the best-performing of the above model types on two held-out test sets:

1. **In-domain:** The 48 held-out samples drawn from the item/task types used for training.
2. **Out-of-domain:** All samples (66) from the two held-out types: Conversations, Respond to a Written Request.

[5] Experiments with the longer guideline-based prompt were unsuccessful: the LLM invariably returns either a commentary on a single testing procedure or rewrites the prompt entirely to handle a single sample.

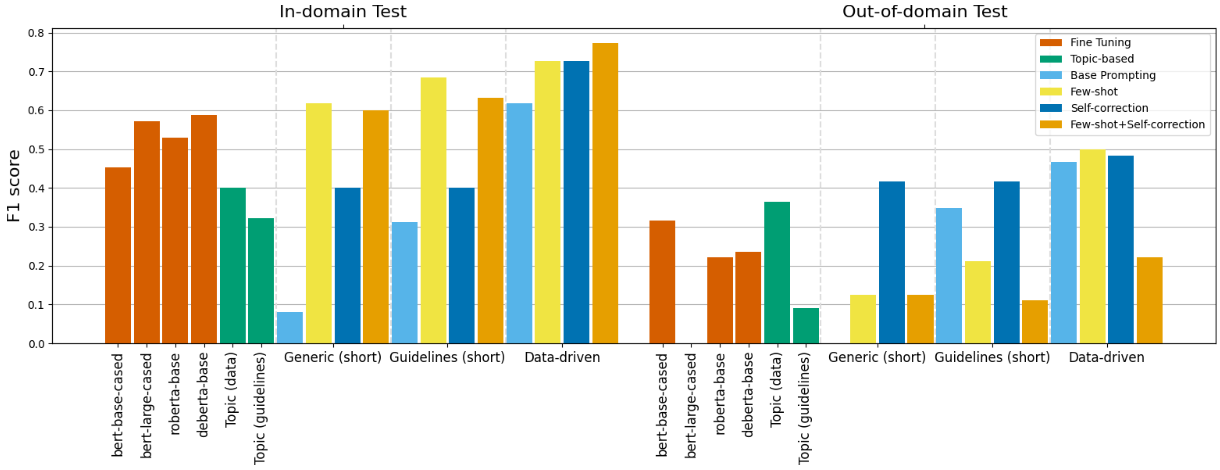


Figure 4: F1 scores on two test sets for each proposed method. Note that for `bert-large-cased` and GENERIC (SHORT), the scores were 0.00 on the unknown test set. Full results in Appendix E.

Figure 4 shows the results on the test set. We evaluate the best-performing models of each type: fine-tuned transformer models, topic-based classification, base prompts, few-shot learning, self-correction, and combining few-shot and self-correction. We here note some key facts about model performance on our test set.

Best Performance Combining the DATA-DRIVEN prompt with self-correction and few-shot learning performs the best on the in-domain test. This shows this is the best approach if there is available data and expertise to support hand-crafting a DATA-DRIVEN prompt and running self-correction. On the out-of-domain data, the smaller initial prompts, GENERIC (SHORT) and GUIDELINE (SHORT) both outperform the DATA-DRIVEN prompt, perhaps due to their more generic nature: the DATA-DRIVEN prompt is too specific to this dataset, and understandably doesn't generalize well. The self-correct+few-shot methodology performs the best in both cases: few-shot learning alone is better than self-correction alone, but the combination is typically the best.

Strong Results from Small Models Traditional transformer-based classification performs remarkably well, especially in generalizing to the out-of-domain data. On the in-domain data, the best performing model `deberta-base` performs on par with the best base prompting model (0.58 compared to 0.60 F1 score), although this is a significant drop from the validation performance of 0.77, and performs quite poorly on out-of-domain data (0.20), indicating the model may overfit during training. On the out-of-domain data, `roberta-base` performs nearly as well as the best-performing overall model, just 0.04 behind the GENERIC (SHORT) prompt with self-correction and few-shot learning. If the goal is to quickly and cheaply build a system that is applicable to a wide variety of domains, there appears to be significant value in relying on these relatively small transformer-based classification models. The Topic (data) approach is also competitive on out-of-domain data, and does not even require model training; it lags only slightly behind the `roberta-base` model.

Self-Correction We found significant success in our proposed self-correction mechanism. While it typically does not outperform few-shot learning in isolation, the methods are naturally complementary, and the combination often yields the best-performing model. In examining the models' self-corrections, we find that when asked to become more restrictive, the model tends to add sentences with new constraints, which nicely reflect the issue that was missed. When asked to become less restrictive, the model tends to add hedges to currently existing constraints.

In our experiments, we noted some issues. First, when run using too many samples or batches, the prompts tend to degrade: once the LLM makes an error and returns a prompt that doesn't match the specifications, the run needs to be aborted. Even when the LLM sticks to the instructions, after many iterations the prompts become unwieldy and self-contradictory, and performance rapidly declines. We suggest using somewhere between six and 20 total samples for prompt self-correction; it is best to avoid making corrections indefinitely.

Model	Type	KSA	Emotion
Fine-tuned	`bert-base-cased`	0.07	0.57
	`bert-large-cased`	0.00	0.00
	`roberta-base`	0.06	0.56
	`deberta-base`	0.08	**0.75**
Topic-based	Data	0.26	0.59
	Guideline-based	0.20	0.06
Base Prompting	GENERIC (SHORT)	0.00	0.00
	GUIDELINE (SHORT)	0.29	0.09
	DATA-DRIVEN	**0.47**	0.50
Self-correction	GENERIC (SHORT)	0.35	0.30
	GUIDELINE (SHORT)	0.35	0.27
	DATA-DRIVEN	**0.47**	0.41
Few-shot	GENERIC (SHORT)	0.18	0.24
	GUIDELINE (SHORT)	0.30	0.24
	DATA-DRIVEN	0.36	0.56
Few-shot + Self-correction	GENERIC (SHORT)	0.18	0.21
	GUIDELINE (SHORT)	0.23	0.21
	DATA-DRIVEN	0.24	0.59

Table 3: Recall scores for KSA and Emotion-labeled data across both test sets.

Use-Cases and Metrics We here report F1 score as a balance between precision and recall. (For full scores, see Appendix E.) Depending on the end use case, other metrics may be more appropriate. In our case, we advocate for always including humans in the evaluation process to ensure that only fair content is accepted. We then value both precision (as we do not want to excessively flag content for fairness issues, which could reduce diversity) and recall (as we do not want to let fairness issues through). Optimizing for recall seems reasonable, as it is likely more important to prevent fairness issues from being released, but it is critical to note that no system is perfect: even optimizing for recall, these fairness issues are likely to persist, and the models should not be used as failproof safeguards.

KSA and Emotion We evaluate performance on the test set for the two subcategories: Knowledge, Skill, and Ability (KSA) and Emotion (Table 3). The `deberta-base` model performs exceptionally well on the KSA subcategory, capturing 75% of the fairness-flagged samples. Data-based methods (the DATA-DRIVEN prompts (0.59) and Topics from Data (0.59)) also perform well, likely due to the inclusion of negative emotional issues in the text. They perform much worse on KSA classification, although the DATA-DRIVEN prompts still yield the best performance (0.47): KSA-related issues are especially difficult as they generally involve only specific knowledge, and would not normally be considered fairness issues in other contexts.

6 Conclusions

This work delivers four key contributions: an exploration of a novel fairness detection task, a dataset of 601 samples annotated for fairness issues, evaluation of a variety of classification models for this task, including fine-tuning, topic-based approaches, and prompting, and a novel prompting strategy, which, combined with few-shot learning, achieves state-of-the-art performance on the task.

This work is aimed to explore the space of fairness and bias issues in generated content, especially in the education context. We aim to highlight the difficulties of accounting for fairness, particularly in specific contexts unlikely to be accounted for by traditional model guardrails. As language model usage becomes more prevalent, the need for proper bias and fairness strategies from people training, deploying, and using these models is paramount.

7 Ethics

Content generation comes with inherent ethical concerns relating to fairness, bias, factuality, and sensitivity. We aim to mitigate these issues regarding fairness, but there are other considerations around generating assessment content. Models may introduce subtle biases against disadvantaged groups, or produce content that appears to be factual, but is not. These are critical failures that need to be accounted for.

In practice, the generation of assessment content requires human intervention: large language model generations are not at the point where they are immune to these negative impacts, and thus for any content that goes into production, a human with relevant expertise needs to evaluate it. The methods we propose support this human intervention, as they can remove obviously offensive content before the human review stage, or assist in human reviews by flagging potentially harmful content.

While our dataset is unlikely to contain any content that is triggering (our framework of fairness is focused on more nuanced contexts), it must be noted that there is potential for it to be used maliciously; for example, by someone designing a system to adapt to and deceive a fairness detection system. In releasing this data, we hope to bring awareness to this issue and better understand the potential negative impacts. Primarily, we stress that any fairness detection system should not be used in isolation or without supervision as a catchall for potential issues.

8 Limitations

Our work is limited largely by the type of content evaluation and the models used. We focus on a small number of item and task types that fall under very specific fairness constraints: the evaluation of the methods used specifically applies to these items under these constraints. This is apparent in the evaluation on the "unseen" item types in Section 5. Applying these methods to new item and task types, even those annotated under the same fairness guidelines, yields significantly reduced results. This is evidence that the methods and models we designed work only for the specific contexts in which they are trained and developed.

Similarly, we explore a small space of models and approaches. We use relatively basic prompt strategies; there exist many other approaches and improvements that are likely to be valuable that we do not evaluate. The same is true of fine-tuned models and topic classification. We present relatively basic, well-known strategies to better understand the difficulty of our data, with the understanding that there are substantial improvements that could be applied.

References

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Perpetual Baffour, Tor Saxberg, and Scott Crossley. 2023. Analyzing bias in large language model solutions for assisted writing feedback tools: Lessons from the feedback prize competition series. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 242–246, Toronto, Canada. Association for Computational Linguistics.

Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. 2018. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943*.

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*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of "bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online. 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. Curran Associates, Inc.

Mayukh Das and Wolf Tilo Balke. 2022. Quantifying bias from decoding techniques in natural language generation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1311–1323, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Sunipa Dev, Emily Sheng, Jieyu Zhao, Aubrie Amstutz, Jiao Sun, Yu Hou, Mattie Sanseverino, Jiin Kim, Akihiro Nishi, Nanyun Peng, and Kai-Wei Chang. 2022. On measures of biases and harms in NLP. In *Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022*, pages 246–267, Online only. Association for Computational Linguistics.

Hannah Devinney, Jenny Björklund, and Henrik Björklund. 2022. Theories of "gender" in nlp bias research. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 2083–2102, New York, NY, USA. Association for Computing Machinery.

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.

Educational Testing Service. 2022. Ets guidelines for developing fair tests and communications.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2023. Promptbreeder: Self-referential self-improvement via prompt evolution. *arXiv preprint arXiv:2309.16797*.

Anjalie Field, Su Lin Blodgett, Zeerak Waseem, and Yulia Tsvetkov. 2021. A survey of race, racism, and anti-racism in NLP. 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 1905–1925, Online. Association for Computational Linguistics.

Hannah Gonzalez, Jiening Li, Helen Jin, Jiaxuan Ren, Hongyu Zhang, Ayotomiwa Akinyele, Adrian Wang, Eleni Miltsakaki, Ryan Baker, and Chris Callison-Burch. 2023. Automatically generated summaries of video lectures may enhance students' learning experience. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 382–393, Toronto, Canada. Association for Computational Linguistics.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

Tanja Heck and Detmar Meurers. 2023. Using learning analytics for adaptive exercise generation. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 44–56, Toronto, Canada. Association for Computational Linguistics.

Saghar Hosseini, Hamid Palangi, and Ahmed Hassan Awadallah. 2023. An empirical study of metrics to measure representational harms in pre-trained language models. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 121–134, Toronto, Canada. Association for Computational Linguistics.

Dirk Hovy and Shrimai Prabhumoye. 2021. Five sources of bias in natural language processing. *Language and Linguistics Compass*, 15(8):e12432.

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, Seattle, Washington. Association for Computational Linguistics.

Alexander Kwako, Yixin Wan, Jieyu Zhao, Mark Hansen, Kai-Wei Chang, and Li Cai. 2023. Does BERT exacerbate gender or L1 biases in automated English speaking assessment? In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 668–681, Toronto, Canada. Association for Computational Linguistics.

Antonio Laverghetta Jr. and John Licato. 2023. Generating better items for cognitive assessments using large language models. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 414–428, Toronto, Canada. 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.

Robert Logan IV, Ivana Balazevic, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. 2022. Cutting down on prompts and parameters: Simple few-shot learning with language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2824–2835, Dublin, Ireland. Association for Computational Linguistics.

Anastassia Loukina, Nitin Madnani, and Klaus Zechner. 2019. The many dimensions of algorithmic fairness in educational applications. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–10, Florence, Italy. Association for Computational Linguistics.

Elijah Mayfield, Michael Madaio, Shrimai Prabhumoye, David Gerritsen, Brittany McLaughlin, Ezekiel Dixon-Román, and Alan W Black. 2019. Equity beyond bias in language technologies for education. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 444–460, Florence, Italy. Association for Computational Linguistics.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2021. StereoSet: Measuring stereotypical bias in pretrained language models. 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 5356–5371, Online. Association for Computational Linguistics.

Praneeth Nemani, Yericherla Deepak Joel, Palla Vijay, and Farhana Ferdouzi Liza. 2024. Gender bias in transformers: A comprehensive review of detection and mitigation strategies. *Natural Language Processing Journal*, 6:100047.

OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Shaina Raza, Deepak John Reji, and Chen Ding. 2024. Dbias: Detecting biases and ensuring fairness in news articles. *International Journal of Data Science and Analytics*, 7:39–59.

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*. Association for Computational Linguistics.

Pedro Saleiro, Benedict Kuester, Loren Hinkson, Jesse London, Abby Stevens, Ari Anisfeld, Kit T. Rodolfa, and Rayid Ghani. 2019. Aequitas: A bias and fairness audit toolkit. *arXiv preprint arXiv:1811.05577*.

Kevin Stowe, Debanjan Ghosh, and Mengxuan Zhao. 2022. Controlled language generation for language learning items. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 294–305, Abu Dhabi, UAE. Association for Computational Linguistics.

Yolande Strengers, Lizhen Qu, Qiongkai Xu, and Jarrod Knibbe. 2020. Adhering, steering, and queering: Treatment of gender in natural language generation. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–14, New York, NY, USA. Association for Computing Machinery.

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 (BEA 2023)*, pages 785–795, Toronto, Canada. Association for Computational Linguistics.

Masaki Uto, Yuto Tomikawa, and Ayaka Suzuki. 2023. Difficulty-controllable neural question generation for reading comprehension using item response theory. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 119–129, Toronto, Canada. Association for Computational Linguistics.

Pranav Narayan Venkit, Sanjana Gautam, Ruchi Panchanadikar, Ting-Hao Huang, and Shomir Wilson. 2023. Unmasking nationality bias: A study of human perception of nationalities in ai-generated articles. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '23, page 554–565, New York, NY, USA. Association for Computing Machinery.

Pranav Narayanan Venkit, Mukund Srinath, and Shomir Wilson. 2022. A study of implicit bias in pretrained language models against people with disabilities. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1324–1332, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Thiemo Wambsganss, Xiaotian Su, Vinitra Swamy, Seyed Neshaei, Roman Rietsche, and Tanja Käser. 2023. Unraveling downstream gender bias from large language models: A study on AI educational writing assistance. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10275–10288, Singapore. Association for Computational Linguistics.

Thiemo Wambsganss, Vinitra Swamy, Roman Rietsche, and Tanja Käser. 2022. 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*, pages 1344–1356, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Yixin Wan, George Pu, Jiao Sun, Aparna Garimella, Kai-Wei Chang, and Nanyun Peng. 2023. "kelly is a warm person, joseph is a role model": Gender biases in LLM-generated reference letters. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3730–3748, Singapore. Association for Computational Linguistics.

Angelina Wang, Xuechunzi Bai, Solon Barocas, and Su Lin Blodgett. 2024. Measuring machine learning harms from stereotypes: requires understanding who is being harmed by which errors in what ways. *arXiv preprint arXiv:2402.04420*.

Hilde Weerts, Miroslav Dudík, Richard Edgar, Adrin Jalali, Roman Lutz, and Michael Madaio. 2023. Fairlearn: Assessing and improving fairness of ai systems. *arXiv preprint arXiv:2303.16626*.

Jakub Wiśniewski and Przemysław Biecek. 2022. fairmodels: a flexible tool for bias detection, visualization, and mitigation in binary classification models. *The R Journal*, 14:227–243.

A Item Types

Table 4 gives examples for each item type.

B Prompts

Below are the five base prompts used. Each also contains a slot to insert a stimulus and a suffix instruction: "Consider the instructions. If the text violates the instructions, return only the word True. If not, return only the word False."

B.1 GENERIC (SHORT)

Ensure it is fair to a test taker. It should not contain biased, toxic, or unethical content. It should not reflect any special skills, abilities, or culture, and it should not contain any negative emotion.

B.2 GENERIC (LONG)

Develop a set of criteria to filter out negative test items that may violate fairness, bias, or sensitivity requirements. Consider the following aspects:

Fairness: Are the test items equally applicable and appropriate for all individuals regardless of their background, demographics, or personal characteristics? Do the test items avoid favoring or disadvantaging any specific group based on race,

Item Type	Example
Read a Text Aloud	Welcome to our house hunting service. Our priority is to help find your dream home. We offer tours of houses with gardens, pool facilities, or spacious garages. However, we also tackle paperwork which can be confusing for first-time buyers. Join us, start your journey towards owning a home.
Talks	Hello, I’m your local council housing officer. I’m reaching out about our new housing construction plans. We are designing affordable, environmentally friendly homes in our area. For your input on these proposed designs, please complete our quick survey. It won’t take more than a few questions. Your opinions are valuable in ensuring these homes meet community needs. Share your thoughts, let’s create a better living environment together.
Text completion	ORG_1 is a locally owned gym that offers a wide range of fitness classes and equipment. We are proud to offer two special deals for our members. The first is a one-month membership for only $50. This includes unlimited access to all our classes and equipment. The second is a three-month membership for $125. This includes a free personal training session and a 10% discount on all additional personal training sessions. Come join us today and take advantage of these great deals!
Respond to Questions Using Information Provided	N: Hello. I received an email about the annual conference that the Association of Professional Journalists is hosting, but I can’t seem to find it. I was hoping you could answer a few questions. Header: [’Annual Small Businesses Conference’, ’Riverside Convention Center’, ’Daily rate: $70] Event table: [”, ’Day 1, 9:00 A.M.’, ’10:30 A.M.’, ’12:00 P.M.’, ’2:00 P.M.’, ’Day 2, 10:30 A.M.’, ’Workshop: Basics of Data Security’, ’Speech: Role of Entrepreneurship’, ’Lunch Break’, ’Presentation: Advanced Data Protection’, ’Speech: Customer Relations and Service’, ’Hans Pham’, ’Hans Pham’, ”, ’Olga Gomez’]
*Conversations	(Woman) Good morning, Alex. Have you reviewed the department’s salary structure for the upcoming year? (Man) Morning, Priya. Yes, I have. There’s a significant disparity, especially when considering the recent promotions. (Woman) It seems like those promoted are receiving disproportionately higher raises. (Man) True. I’ll propose a more balanced salary increase for all levels at the next management meeting. It should help in retaining our skilled employees.
*Respond to a Written Request	date: [’January 8’] signature_line: [’Best regards, Shani Flores, Airline Customer Support Specialist’] from: [’Shani Flores, Airline Customer Support Specialist’] to: [’Hassan Kumar’] body: [’Thank you for choosing our airline for your recent travel.Your feedback is valuable. Could you share your thoughts on what we could do better during the flight?’] subject: [’Improving in-flight services’]

Table 4: Item types and annotations for fairness issues. Each has a binary annotation for containing a fairness issue and then can be also tagged as containing a KSA issue or an Emotion issue. Multiple subcategories are allowed. *Items are held out for testing, and not used for any training/validation.

ethnicity, gender, religion, sexual orientation, socioeconomic status, disability, or any other protected characteristic?

Bias: Do the test items avoid stereotypes or assumptions that may lead to unfair advantages or disadvantages for certain groups? Are the language, examples, scenarios, and references used in the test items neutral and inclusive? Have the test items been reviewed to ensure they do not unintentionally favor or discriminate against any particular group?

Sensitivity: Are the test items respectful and considerate of individuals' diverse experiences, perspectives, and identities? Do the test items avoid potentially triggering or offensive content?

B.3 GUIDELINE (SHORT)

Consider the following text. The text should avoid any content or information that would unfairly disadvantage someone tested on it. It should not do any of the following:

1. It should not contain content specific to a region, including brand names, celebrities, sports, entertainment, culture, customs, geography, government, history, holidays, or laws. Any of these topics that are specific to a certain region should be excluded.

2. It should not include topics that require specialized knowledge that would be easier for one group than another.

3. It should not include unnecessarily difficult language including words, phrases, and concepts more likely to be known by one group than another.

4. It should avoid contexts that not all people may have experienced. The contexts should not require direct, personal experience to understand.

5. It should not mention religion.

6. It should not include contemptuous, derogatory, or exclusionary language. It should not induce any negative emotions.

7. It should not advocate for particular causes or ideologies, or include anything divisive.

8. It should avoid sensitive and controversial topics, including political issues, natural disasters, accidents, or other negative topics.

B.4 GUIDELINE (LONG)

Below are a set of guidelines. These guidelines aim to enhance the fairness and validity of tests, communications, and other materials. These guidelines assist users in understanding fairness in assessment, including the right content, eliminating unfair content, promoting diversity and inclusivity, addressing accessibility and inclusion issues and reducing subjective fairness decisions. The guidelines cover the fairness of various subjects including the National Assessment of Educational Progress (NAEP), K–12 tests, artificial intelligence (AI) algorithms and includes information to help use plain language and a quick reference guideline list.

Understanding fairness in testing is crucial for proper application of guidelines, though its definition varies. One common definition sees fairness as absence of any inequity, affecting individuals and groups alike, such as unfair test questions or biased content affecting diverse groups. Another definition argues that tests seeming harder for certain groups aren't necessarily unfair, as differences in results may reflect real differences in knowledge or ability, not test bias. Group score differences don't prove bias, but should be explored to rule out bias. Furthermore, fairness definitions based on outcomes are contested and of limited use during test design. Fairness is also defined based on test validity. The test validity indicates quality, and represents the accuracy of inferences and actions based on scores, which must be equally valid for all test-takers for a test to be fair. Therefore, an effective definition of fairness in assessment is rooted in validity, creating an interconnected relationship between the two. Lastly, fairness in testing relates to the effectiveness of related educational products and services in fulfilling their intended purposes.

These guidelines should ideally cater to everyone, particularly focusing on groups discriminated against due to factors such as age, appearance, citizenship, disability, ethnicity, gender, national origin, native language, race, religion, sexual orientation, and socioeconomic status. It's crucial to also account for intersectionality, a framework recognizing how overlapping identities like race and gender can impact the experiences of individuals with multiple marginalized identities. For instance, Black women may perceive test material differently than Black men or White women.

Principles and Guidelines for Fairness

Fairness in assessment requires adherence to key principles: Tests should focus on essential aspects of the intended construct and avoid construct-irrelevant hurdles. They must offer design, content, and conditions facilitating valid inferences about diverse test takers' knowledge and abilities. Also, they should provide scores that allow valid groupwise inferences. The subsequent sections offer specific guidelines related to these principles. In case of interpretational conflicts, choose the one that upholds fairness principles.

Construct-Irrelevant KSA Barriers to Success

Construct-irrelevant Knowledge, Skills, and Abilities (KSA) barriers to test success can arise when unrelated KSAs are required to answer a question correctly. For example, a math item asking for the conversion of kilometers to meters is construct-irrelevant to multiplication skills. If a specific group lacks this irrelevant knowledge, the test's validity and fairness are diminished. Construct-irrelevant sources of KSA often include unfamiliar contexts, disabilities, difficult language, regionalisms, religion, specialized knowledge, translation issues, unfamiliar item types, and topics specific to the U.S.

The content and context of test stimuli should be familiar and accessible to all test takers. Tests shouldn't require personal experiences that may not be available to test takers with disabilities.

Language should be simple and clear, and shouldn't require knowledge of jargon or specialized vocabulary unless relevant to the test.

Regionalisms, words or phrases specific to a certain region, should not be required unless relevant to the test construct.

Tests shouldn't require unrelated knowledge about religion.

Construct-irrelevant specialized knowledge should be avoided unless the test is designed to assess that specific knowledge.

Tests need to be culturally adapted along with translations to ensure fairness.

Test takers should be familiar with the technology used in assessments.

Tests taken by an international audience shouldn't require specific knowledge of U.S. dominant cultures or conventions unless meant to measure such knowledge.

Construct-Irrelevant Emotional Barriers to Success

Construct-irrelevant emotional barriers to success occur during testing when certain language, scenarios or images elicit strong emotions that disrupt a test taker's ability to answer a question. This can happen due to offensive content, controversial material, or content that challenges a test taker's personal beliefs. The stress and pressure of testing can heighten these reactions. It's important to avoid potentially offensive material, especially content that may trigger negative reactions in diverse groups of test takers.

Test content about groups that have been discriminated against should be carefully reviewed for any offensive or emotionally triggering material. Test developers should strive for diversity in their team and aim to use content written by diverse authors. However, offensive content should be avoided even in multiple choice items where the wrong answer may potentially be seen as the viewpoint of the test creators or institution.

A list of topics likely to trigger negative reactions is provided, including topics like abduction, abortion, and drug use among others, and should be avoided in test materials unless they are crucial for test validity. On the other hand, while some topics may not trigger negative reactions, they need careful handling to ensure balance and objectivity. This includes topics like advocacy, biographical material, conflicts and others.

The document concludes with a detailed discussion on specific topics that should either be avoided or handled with care in tests, including religion, personal questions, role playing, sexual behavior, stereotypes, and violence among others. It emphasizes the importance of fair, balanced and objective representation in testing material, and the avoidance of content that may trigger strong negative emotions or construct-irrelevant barriers to test performance.

Plain Language

Tests should contain plain language. Plain language benefits all test-takers, minimizing score differences unrelated to test construct. It is not designed to override client-specific guidelines or simplify complex language inherent to the construct being tested. Plain language applies to all irrelevant elements of tests and associated materials, and examples where it isn't suitable include reading comprehension tests, subject-matter tests, historical documents, and language proficiency assessments.

Plain language includes writing short, clear paragraphs with one main idea and avoiding complex sentences and vocabulary unless necessary. Relative clauses should be used sparingly, nouns should be clear, transition words used for clarity, and personal pronouns used when aiding communication. Testing material layout and formatting should be clear, and verb forms simple and in active voice. Test-specific guidelines also apply, with clear and precise stem presentation. Avoiding negative stems or emphasizing them if necessary is advised.

B.5 DATA-DRIVEN

No mention of anything difficult or challenging.

Nothing about luxury or elitist activities such as cruises or golf, or expensive vacations.

Nothing upsetting or dangerous such as car accidents, health hazards, disease, or disasters that might make the test taker nervous.

No topics that might be anxiety-inducing, like romance, testing, or relationships.

No health-related topics like health codes, health hazards, or anything negative related to health. General well-being is okay.

No fitness discussion that is focused on weight loss or negative health impacts.

No negativity involving overcrowding, housing shortages, or general bad living conditions or housing concerns.

No content and vocabulary specific to the US or other regions such as Fahrenheit units, local holidays, or local events.

No controversial and political topics such as fake news.

No mentions of any alcohol, drugs, or other controlled substances.

No specific skills or abilities that wouldn't be generally known, such as law or medicine. No specialized vocabulary, acronyms, or jargon.

No hazardous or potentially dangerous technology like viruses or artificial intelligence.

No trademarked or well-known products, places, or people.

C Self-Correction Prompts

Below is an example of prompt self-correction:

Initial Prompt Consider the following text. Ensure it is fair for a test taker. It should not contain biased, toxic, or unethical content. It should not reflect any special skills, abilities, or culture, and it should not contain any negative emotion.

After one batch of corrections Ensure the test content is unbiased, non-toxic, and ethical, without favoring any specific skills, hobbies, or activities not related to the test's subject. It should not evoke negative emotions or assume uniform experiences or opportunities among test takers. The content should avoid asking about personal financial decisions or preferences. Questions that ask for personal opinions or experiences are acceptable as long as they do not favor a specific group or assume uniform experiences.

After a final batch of corrections Ensure the test content is impartial and ethical, not favoring unrelated skills or activities. It should not assume similar experiences among test takers or provoke negative emotions. Avoid questions about personal finances, specific hobbies, or activities that may not be universally accessible or common. Personal opinion or experience questions are acceptable if they don't favor a certain group and are not related to sensitive personal information. Also, avoid questions that assume a certain life stage or financial status, such as retirement planning, as not all test takers may have the same experiences or opportunities. Return 'True' if these principles are breached, 'False' otherwise.

D Validation Results

Precision, recall, and F1 scores for each model on the validation set can be found in Table 5.

E Test Results

Precision, recall, and F1 scores for all models on both test sets can be found in Table 6.

D

Base Prompting

Prompt	Prec	Rec	F1
Generic (short)	0.00	0.00	0.00
Generic (long)	0.00	0.00	0.00
Guidelines (short)	0.67	0.25	0.36
Guidelines (long)	1.00	0.04	0.08
Data-driven	0.88	0.58	0.70

Few-shot Prompting

Prompt	n	Prec	Rec	F1
Generic (short)	3	0.92	0.46	0.61
	5	0.92	0.46	0.61
Guidelines (short)	3	0.88	0.58	0.70
	5	0.88	0.58	0.70
Data-driven	3	0.81	0.71	0.77
	5	0.89	0.67	0.76

Self-Correction

Prompt	Prec	Rec	F1
Generic+correction	0.50	0.29	0.37
Guidelines (short)+correction	0.90	0.38	0.53
Data-driven+correction	0.85	0.71	0.77

Combining Few-Shot and Self-Correction

Prompt	n	Prec	Rec	F1
Generic+correction	3	0.81	0.54	0.65
Generic+correction	5	0.79	0.63	0.70
Guideline+correction	3	0.88	0.58	0.70
Guideline+correction	5	0.88	0.58	0.70
Data-driven+correction	3	0.90	0.75	0.82
Data-driven+correction	5	0.90	0.75	0.82

Table 5: Results on the validation set for all prompting strategies.

Method	Details	Test (Known)			Test (Unknown)		
		Prec	**Rec**	**F1**	**Prec**	**Rec**	**F1**
Fine-tuning	`bert-base-cased`	1.00	0.29	0.45	0.75	0.20	0.32
	`bert-large-cased`	0.91	0.42	0.57	0.00	0.00	0.00
	`roberta-large`	0.90	0.38	0.53	0.67	0.13	0.22
	`deberta-base`	1.00	0.42	0.58	1.00	0.13	0.24
Topic-based	Topics from Data	0.64	0.29	0.40	0.57	0.27	0.36
	Topics from Guidelines	0.71	0.21	0.32	0.14	0.07	0.09
Base prompting	Generic (short)	1.00	0.04	0.08	0.00	0.00	0.00
	Guidelines (short)	0.63	0.21	0.31	0.50	0.27	0.35
	Data-driven	0.72	0.54	0.62	0.47	0.47	0.47
Few-shot $n = 3$	Generic (short)	0.72	0.54	0.62	1.00	0.67	0.13
	Guidelines (short)	0.93	0.54	0.68	0.50	0.13	0.21
	Data-driven	0.80	0.67	0.73	1.00	0.33	0.50
Self-correction	Generic (short)+correction	0.50	0.33	0.40	0.56	0.33	0.42
	Guideline (short)+correction	0.64	0.29	0.40	0.50	0.47	0.48
	Data-driven+correction	0.82	0.38	0.51	0.56	0.33	0.42
Self-correction + few-shot $n = 3$	Generic+correction	0.75	0.50	0.60	1.00	0.07	0.13
	Guideline (short)+correction	0.86	0.50	0.63	0.33	0.07	0.11
	Data-driven+correction	0.85	0.71	0.77	0.67	0.13	0.22

Table 6: Results for each of our methods on the two held-out test sets.

Towards Automated Document Revision: Grammatical Error Correction, Fluency Edits, and Beyond

Masato Mita[1,2] **Keisuke Sakaguchi**[3] **Masato Hagiwara**[4,5]
Tomoya Mizumoto[2] **Jun Suzuki**[3,2] **Kentaro Inui**[6,3,2]
[1]CyberAgent [2]RIKEN AIP [3]Tohoku University
[4]Earth Species Project [5]Octanove Labs [6]MBZUAI

Abstract

Natural language processing (NLP) technology has rapidly improved automated grammatical error correction (GEC) tasks, and the GEC community has begun to explore *document-level* revision. However, there are two major obstacles to going beyond automated *sentence-level* GEC to NLP-based *document-level* revision support: (1) there are few public corpora with document-level revisions annotated by professional editors, and (2) it is infeasible to obtain all possible references and evaluate revision quality using such references because there are infinite revision possibilities. To address these challenges, this paper proposes a new document revision corpus, **Te**xt **R**evision of **A**CL papers (TETRA), in which multiple professional editors have revised academic papers sampled from the ACL anthology. This corpus enables us to focus on document-level and paragraph-level edits, such as edits related to coherence and consistency. Additionally, as a case study using the TETRA corpus, we investigate reference-less and interpretable methods for meta-evaluation to detect quality improvements according to document revisions. We show the uniqueness of TETRA compared with existing document revision corpora and demonstrate that a fine-tuned pre-trained language model can discriminate the quality of documents after revision even when the difference is subtle.

1 Introduction

Document revision is a crucial step in the process of writing essays and argumentative texts. The writing process consists of two major parts: content organization and selection planning (henceforth, *planning part*) and realization of text improvement (henceforth, *realization part*), which are hierarchical and recursive. In addition, according to previous studies on argumentative writing (Flower and Hayes, 1981; Beason, 1993; Buchman et al., 2000; Seow, 2002; Allal et al., 2004), *realization part* in writing process typically comprises three main stages: *Revising*, *Editing*, and *Proofreading*. *Revising* is the initial editing step used to plan and structure the overall document at a high level, *Editing* focuses on making sentence-level or phrase-level expressions, and *Proofreading* is used to identify and correct errors such as spelling and grammar errors (see Figure 1, left). While the order of these steps is not set in stone, the writing process typically starts with a broad, high-level perspective, and gradually narrows down the scope of edits.

In contrast to the typical human writing process, GEC research in NLP field, which is primarily intended to support writing, initially focused on a fine-grained scope, e.g., spelling errors (Brill and Moore, 2000; Toutanova and Moore, 2002; Islam and Inkpen, 2009) and closed-class parts of speech (such as prepositions and determiners) (Han et al., 2006; Nagata et al., 2006; Felice and Pulman, 2008). The research community then expanded its focus to include edits at the phrase and sentence levels while also considering fluency (Sakaguchi et al., 2016; Napoles et al., 2017) (Figure 1, right). However, significantly less work has been done on *document-level* revisions due to two major challenges. First, document revisions encompass a broader range of concerns such as coherence and flow, compared to conventional GEC and fluency correction, which makes it difficult to find publicly available corpora that have been annotated by experts (professional editors). Second, evaluating the quality of revisions is challenging as it requires multiple reference points, as there are many ways to revise a single document. This suggests that *reference-less* evaluation metrics (Napoles et al., 2016; Choshen and Abend, 2018; Islam and Magnani, 2021) are hold significant importance in automated document revision models.

Considering these challenges associated with automated document revision, we propose a new high-quality corpus and explore possibilities for transpar-

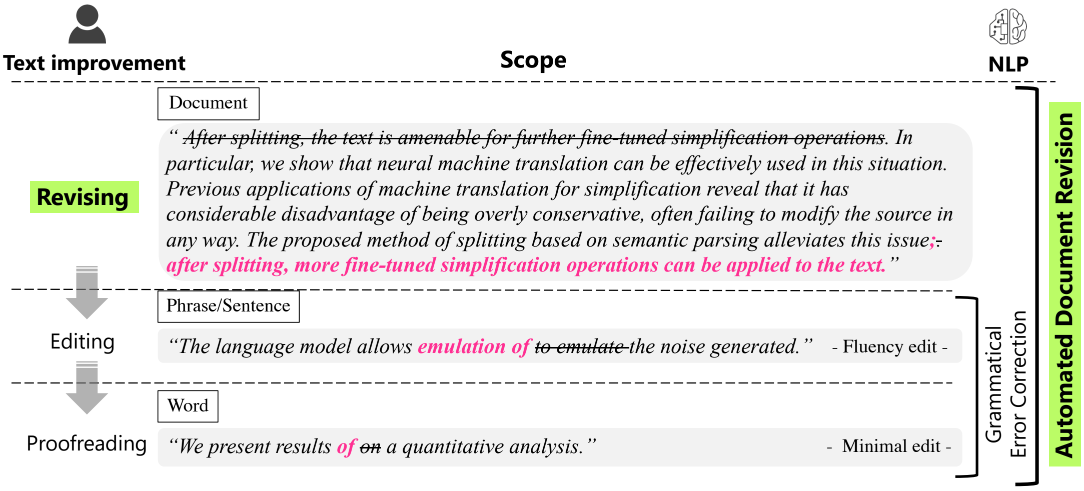

Text improvement
Scope
NLP
Document
" ~~After splitting, the text is amenable for further fine-tuned simplification operations~~. In particular, we show that neural machine translation can be effectively used in this situation. Previous applications of machine translation for simplification reveal that it has considerable disadvantage of being overly conservative, often failing to modify the source in any way. The proposed method of splitting based on semantic parsing alleviates this issue;~~.~~ after splitting, more fine-tuned simplification operations can be applied to the text."
Revising
Editing
Phrase/Sentence
"The language model allows emulation of ~~to emulate~~ the noise generated." - Fluency edit -
Proofreading
Word
"We present results of ~~on~~ a quantitative analysis." - Minimal edit -
Grammatical Error Correction
Automated Document Revision

Figure 1: Overview of the scope for automated document revision. Each example is taken from TETRA corpus. We focus on the document revision process which has been overlooked by GEC. Automated document revision extends the scope of GEC.

ent evaluation methods that are independent of gold standards or references. Our corpus, **T**ext **R**evision of **A**CL papers (TETRA), comprises academic papers from the ACL anthology with document-level revisions, revision types, and concrete feedback comments annotated by multiple professional editors. This corpus was designed based on a new XML-based annotation scheme that can handle edit types beyond sentences (e.g., argument flow) in addition to conventional word-level and phrase-level edits. TETRA has uniqueness in terms of the number of references, the expertise level of the editors, and topic diversity.

As a case study, we use TETRA to investigate whether it is possible to build an **i**nstance-wise **r**evision **c**lassification (IRC) method, in which a model can distinguish pre-edited or post-edited versions for a given single revision pair. In recent years, several studies have been conducted on the use of large language models (LLMs) as evaluators in language generation tasks. For example, GPT-4 (OpenAI, 2023) has demonstrated superior performance compared to existing automatic evaluation metrics in text summarization, dialogue generation, and machine translation (Liu et al., 2023; Kocmi and Federmann, 2023). In light of this current situation, we conduct experiments to evaluate how well pre-trained language models, such as BERT (Devlin et al., 2019) and LLMs such as GPT-4, can perform as a (meta-)evaluation method for each edit type, both with and without fine-tuning. The results demonstrate that the supervised method can accurately choose post-edited snippets with an accuracy of 0.85 to 0.96, indicating the feasible potential of automated evaluation in document revision.

We release TETRA to the public, and hope that it will encourage the community to work towards automated document-level revision.[1]

2 Background

The field of GEC, which has a multi-decade history, began with the goal of detecting and correcting targeted error types and providing feedback to English as a second language learners.[2] Early GEC systems primarily focused on a limited number of closed-class error types, such as articles (Han et al., 2006) and prepositions (Chodorow et al., 2007; Tetreault and Chodorow, 2008; Tetreault et al., 2010; Cahill et al., 2013; Nagata et al., 2014). The scope of GEC was later expanded to include all types of errors, including verb forms, subject-verb agreement, and word choice errors (Lee and Seneff, 2008; Tajiri et al., 2012; Rozovskaya and Roth, 2014). This line of research led to the establishment of shared benchmark tasks (Dale and Kilgarriff, 2011; Dale et al., 2012; Ng et al., 2013, 2014).

Motivated by the observation that error-coded local edits do not always sound natural to native speakers, the scope of GEC has been further expanded from word-level closed-class edits to phrase-level and sentence-level *fluency* ed-

[1] `https://github.com/chemicaltree/tetra`

[2] In this paper, we focus on GEC literature after the 2000s when statistical were widely adopted. For a comprehensive history of GEC in the 1980s and 1990s, including rule-based approaches, please refer to Leacock et al. (2014).

🟥 Grammaticality 🟪 Fluency 🟨 Clarity 🟩 Style 🟧 Readability 🟫 Redundancy 🟦 Consistency
This paper presents empirical studies and closely corresponding theoretical models of a chart parser's performance while~~the performance of a chart parser~~ exhaustively parsing the Penn Treebank with the Treebank's own context-free grammar (CFG)~~CFG grammar~~. We show how performance is dramatically affected by rule representation and tree transformations, but little by top-down vs. bottom-up strategies. We discuss grammatical saturation, provide an~~, including~~ analysis of the strongly connected components of the phrasal nonterminals in the Treebank, and model how, as sentence length increases, regions of the grammar are unlocked, increasing the effective grammar rule size ~~increases as regions of the grammar are unlocked,~~ and yielding super-cubic observed time behavior in some configurations.
We expect this approach to yield the following three improvements. Taking advantage of the representation learned by the English model will lead to shorter training times compared to training from scratch. Relatedly, the model trained using transfer learning will require ~~requires~~ less data for an equivalent score than a German-only model. Finally, the more layers we freeze the fewer layers we will need to back-propagate through during training; thus,~~. Thus~~ we expect to see a decrease in GPU memory usage since we do not have to maintain gradients for all layers.
We present the results of ~~on~~ a quantitative analysis of a number of publications in the NLP domain on the collection~~collecting~~, publishing, and availability of research data. We find that, although a wide range of publications rely on data crawled from the web, ~~but~~ few publications provide~~give~~ details of~~on~~ how potentially sensitive data was treated. In addition ~~Additionally, we find that~~, while links to repositories of data are given, they often do not work, even a short time after publication. We present~~put together~~ several suggestions on how to improve this situation based on publications from the NLP domain, as well as ~~but~~ also other research areas.

Table 1: Examples of revision. Each edit type is highlighted respectively.

its (Sakaguchi et al., 2016). With this expansion, the community has proposed new benchmark datasets (Daudaravicius et al., 2016; Napoles et al., 2017; Bryant et al., 2019; Napoles et al., 2019; Flachs et al., 2020; Zhang et al., 2023) and evaluation metrics (Dahlmeier and Ng, 2012; Felice and Briscoe, 2015; Napoles et al., 2015; Bryant et al., 2017; Napoles et al., 2019; Gotou et al., 2020; Gong et al., 2022; Ye et al., 2023) for sentence-to-sentence GEC. In addition, GEC models with deep neural network (DNN) techniques have been developed. Such models are robust against word-level and phrase-level local edits in a given sentence and exhibit human-parity performance on some benchmark datasets (Yuan and Briscoe, 2016; Ji et al., 2017; Chollampatt and Ng, 2018; Ge et al., 2018; Kiyono et al., 2019; Kaneko et al., 2020; Rothe et al., 2021; Li et al., 2023; Yang et al., 2023; Fang et al., 2023; Cao et al., 2023).

In contrast to the significant advancements in the area of grammar and fluency correction, relatively few studies have explored revisions for *document-level argumentative writing*, which require a greater investment of time and resources to create appropriate corpora or datasets. Lee and Webster (2012) made an initial attempt to construct a document revision corpus comprising 13,000 student writings with feedback comments from tutors in the Teaching English to Speakers of Other Languages (TESOL) program. Although the authors developed labels for paragraph-level revisions (e.g., coherence), only 3% of all revisions were annotated as paragraph-level revisions, 90% of the revisions were at the word-level, and 7% were at the sentence-level. This is because the corpus comprises writing from language learners, and the majority of errors were simple grammar and fluency errors. This lesson highlights the importance of using a corpus for document-level revision that has already been partially edited for grammar and fluency. However, due to copyright restrictions, this corpus may not be publicly available. The data source for a document-level corpus should be openly licensed to encourage community-based open research in the long term.

Another line of work (Zhang and Litman, 2014, 2015; Zhang et al., 2016, 2017; Kashefi et al., 2022) has created the ArgRewrite corpus, a collection of 86 argumentative essays that include three drafts, each with two cycles of revisions, and edit labels. The ArgRewrite corpus (both v1 and v2) contains roughly half of all edits as surface-level corrections (e.g., conventional GEC or fluency edits), and the other half of edits as content-level document revisions. While the ArgRewrite corpus has more document-level revisions than the corpus of Lee and Webster (2012), all of the essays in the ArgRewrite corpus were written on the same topic. The first version of the ArgRewrite corpus (Zhang et al., 2017) discusses the topic of *whether the proliferation of electronic enriches or hinders the development of interpersonal relationships*, and the second version (Kashefi et al., 2022) focuses on *whether to support or against self-driving cars*.

	Lee and Webster (2012)	Zhang et al. (2017)	Kashefi et al. (2022)	Du et al. (2022)	Ours (TETRA)
# docs	3,760	60	86	559	64
# sents (avg)	-	18.7	25.8	7.19	**26.92**
# references	1	1	1	1	**3**
Edit scope	Form?	Content&Form	Content&Form	Content&Form	**Form**
% beyondGECs	3.2	49.4	52.6	52.8	**56.9**
Drafted by	ESL	Native (*ESL)	Native (*ESL)	Native (*ESL)	**ESL/Native**
Revised by	Author (NonExp.)	Author (NonExp.)	Author (NonExp.)	Author (NonExp.)	**Exp.**
Edit-types by	NonExp.	NonExp.	NonExp.	NonExp.	**Exp.**
Feedback					✓
Topic diversity	✓			✓	✓
Public availability		✓	✓	✓	✓

Table 2: Characteristics of TETRA corpus compared to existing document revision corpora. The **uniqueness of TETRA** is highlighted. *Exp.* and *NonExp.* means expert and non-expert, respectively. *Edit scope* indicates whether it includes edits regarding content and/or form. *% beyondGECs* shows the ratio of edits that are not covered by GEC edit types. *Drafted by* indicates who wrote the (first) draft, *Revised by* shows who revised the draft, *Edit-types by* shows who annotates edit types. *Feedback* (✓) presents whether the corpus contains feedback comments or not. *Topic diversity* (✓) presents whether the corpus contains two or more topics, or a single topic only (no ✓). *Public availability* (✓) shows whether the corpus is publicly available to the community. Native (*ESL) indicates that most of the documents are drafted by native speakers, but some ESL is included.

This lack of topic diversity can lead to overfitting when developing and evaluating automated document revision models (Mita et al., 2019).

Recently, Du et al. (2022) released a corpus of iterative document revisions from Wikipedia, arXiv, and Wikinews, with edit intention labels annotated[3]. Although this work shares the same objective as ours, there are some differences such as the revision scope, the number of references, the expertise level of the editors, and the absence of feedback comments (Table 2). Furthermore, their annotations are done at a sentence level, whereas our dataset (TETRA) is annotated at a document (and sentence) level. Therefore, our dataset (TETRA) complements their corpus (and vice versa).

3 Automated Document Revision

Given a source document d that consists of paragraphs, a potentially automated editor f revises (R) d into d' ($f : d \mapsto d'$). Here, revision R is a set of edits e, and an edit e is defined as a tuple $e = (\mathit{src}, \mathit{tgt}, t, c)$, where *src* is the source phrase before the revision, *tgt* is the revised phrase, t is the edit type (e.g., grammar, word choice, or consistency), and c represents (optional) rational comments about the edit. When *src* is empty (Ø), this edit indicates *insertion*, and it indicates *deletion* when *tgt* is empty; otherwise, the edit is considered to be a *substitution*. Automated document revision includes various edit types (t), e.g., mechanics, word choice, conciseness, and coherence. This is discussed in further detail in §4.4. Note that t does not exclude the scope of conventional (sentential and subsentential) grammatical error and fluency correction. Rationale comments (c) are a useful resource in the study of feedback generation, which has become prominent in the GEC community (Nagata, 2019; Hanawa et al., 2021; Nagata et al., 2021). Thus, automated document revision is a natural extension of sentence-level error correction to document-level error correction with a wider context.

4 The TETRA Corpus

The validity of a dataset design is contingent upon the purpose and goals of the study. In line with §1 (and also Figure 1), the primary objective of this study is to introduce a novel task focused on enhancing document-level editing and its automated evaluation technologies, which is distinct from the existing GEC task. It is important to note that our aim is not to contribute to a broader understanding of “human revision” in general, which sets our study apart from the previous studies on revision (mentioned in §2).. Hence, it is crucial to create a dataset that minimizes the inclusion of minor grammatical errors and fluency-related edits, which are already emphasized as requirements in GEC. This is essential because proposing a new task entails the need to distinguish the technological aspects and linguistic phenomena targeted by the existing task and the proposed task.

[3] We are aware that other subsequent studies (Jiang et al., 2022; D’Arcy et al., 2023) and on text revision have appeared since the preprint of this study was published.

Aspects	Edit types (abr.)	Definition	Scope	%
Grammaticality	grammar, capitalization	edits that aimed to fix spelling/grammar mistakes	S	19.4
Fluency	word choice, word order	edits that aimed to increase sentence fluency	S	23.7
Clarity	clarity	edits that aimed to amplify meaning for clarity	S/D	19.4
Style	style, tone	edits that aimed to adapt the style	S/D	8.0
Readability	readability	edits that aimed to improve readability	S/D	16.8
Redundancy	redundancy, conciseness	edits that aimed to reduce redundancy	S/D	7.2
Consistency	consistency, flow	edits that aimed to increase paragraph fluency	D	5.5

Table 3: Definition of edit types. S and D (in the *scope* column) indicate the sentence and the document, respectively. We highlight edit types that rely on beyond sentence-level context to edit.

4.1 Data Source

To meet the aforementioned requirement, we utilized the ACL anthology [4] papers as our source data. These papers are generally well-written, peer-reviewed papers on NLP. This choice was made based on the hypothesis that addressing minor errors, such as grammatical errors, is necessary to observe global edits that improve coherence and consistency. Furthermore, (2) we chose the abstract and introduction sections since these sections tend to contain fewer embedded math and complex citations than other sections , and they are more likely to induce global editing specific to the document level due to their greater linguistic freedom.

We selected the source documents from the ACL anthology as follows. First, we created eight groups (=2^3) based on the possible combinations of three different attributes: (1) whether the paper was published at a conference or a workshop, (2) whether the paper is affiliated with a native vs. non-native English speaking country, and (3) whether the first author was a student (at the time the paper was published). We randomly sampled papers until we obtained eight unique papers for each group (i.e., 64 papers in total).

4.2 Annotation Scheme

The scope and granularity of edit types vary widely in previous studies, and there is no standard set of labels. Thus, we define categories of edit types (Table 3) based on previous literature on argumentative and discourse writing (Kneupper, 1978; Faigley and Witte, 1981; Burstein et al., 2003; Zhang et al., 2017). Table 1 provides concrete examples of each type of edit in TETRA.

To create the proposed TETRA, we selected an XML format for the following reasons. First, XML is easy to parse using standard libraries (e.g., Python ElementTree and the Java DOM parser)[5] compared to other formats that frequently require exclusive scripts. Such exclusive scripts incur higher maintenance costs to keep up with the updates of additional dependencies. Second, XML is more flexible than other formats in terms of embedding additional information, such as edit types, edit rationale, comments, and other meta information. For example, as shown in Table 1, document revisions include edit types based on various evaluation aspects, and can be further annotated for each edit with their rational comments using a flexible XML scheme (See Appendix C). Furthermore, edits beyond a single sentence, including sentence merging, splitting, and reordering, can be annotated in a flexible manner (See lines 5-7 in Table 7).

4.3 Annotators

We recruited three professional editors with years of experience editing and proofreading English academic writing, who are native English speakers, to independently revise all 64 documents on the Google Docs platform. They added an edit rationale whenever appropriate, and the revised documents were converted to XML format by the first two authors.[6] Information on how to recruit annotators and instructions for them can be found in the Appendix A and B, respectively.

4.4 Statistical Analysis

Table 2 summarizes the characteristics of TETRA corpus compared to existing document revision corpora. We can first emphasize the quality of the TETRA corpus since it is the only document

[4] https://aclanthology.org

[5] We made the nest of XML tags as shallow as possible for users to parse documents even more easily. In TETRA, the maximum depth of nested XML tags is two. We have established an annotation policy for cases of intersecting edit spans, but we did not encounter any such cases made by professional editors.

[6] During the conversion process, minor corrections and remapping of edit types were made only as necessary.

	Student		Non-student		Native		Non-native		Conf.		WS	
Aspects	#	%	#	%	#	%	#	%	#	%	#	%
Grammaticality	79	19.5	106	21.5	60	16.5	125	21.3	110	22.7	75	16.2
Fluency	115	25.2	110	22.4	74	20.4	151	25.8	99	20.4	126	27
Clarity	100	21.9	84	17.1	88	24.2	96	16.4	84	17.3	100	21.6
Style	39	8.5	37	7.5	29	8.0	47	8.0	46	9.5	30	6.5
Readability	74	16.2	85	17.3	75	20.7	84	14.3	92	19.0	67	14.4
Redundancy	32	7.0	36	7.3	22	6.1	46	7.8	25	5.2	43	9.3
Consistency	18	3.9	34	6.9	15	4.1	37	6.3	29	6.0	23	5.0

Table 4: Distributions of revision aspects by writer's attributes.

Levels	Avg	Min	Max
detection	0.32	0.27	0.35
correction	0.83	0.75	1.00

Table 5: Two levels of inter-annotator agreement: agreement on *detection* and *correction*.

revision corpus that is annotated with revisions by multiple experts, whereas most existing document revision corpora are based on revisions by authors themselves, leaving the quality of revisions in doubt. Existing corpora also have the limitation that the editor (*Revised by*) and the edit type annotator (*Edit-type by*) do not coincide, and thus cannot fully reflect the edit intent, but TETRA corpus overcomes this limitation since the edit type is provided by the person who made the revision. Furthermore, we find that the TETRA corpus contains more edits beyond the GEC (*% beyondGECs*) than the existing corpora, indicating that our hypothesis in source data selection (§4.1) is valid.

The right-most column in Table 3 shows the distribution of edit types found in 16 randomly sampled papers (i.e., 25% of the proposed TETRA corpus). We found that 56.9% of the edits were related to issues beyond the sentence-level context (e.g., redundancy), which is greater than other document revision corpora (Table 2). This is simply because TETRA's source documents are academic papers that have already been proofread to some degree compared to other existing document revision corpora where language learner essays are used as the source material. In terms of the differences among the three different attributes (§ 4.1), we did not find any clear trends, which indicates that the quality of papers in the ACL corpus is uniformly good across the venue and author attributes. The details are shown Table 4.

In document-level revision, it is not straightforward to compute inter-annotator agreement due to the diversity of potential revisions and the broad scope of applicable edits. Thus, we measured two levels of inter-annotator agreement, i.e., (1) agreement on *detection* and (2) agreement on *correction*. The first measurement computes how frequently edit spans overlap (i.e., agree) among annotators, and the second measurement computes how frequently edit type labels (e.g., clarity) match when two or more annotators detect the same (or overlapped) span. Table 5 shows the results.

The result demonstrates that the expert annotators agreed on the direction of editing when they decided an issue was in a certain span (the agreement rate on *correction* was approximately 0.8); however, the experts disagreed on where to consider an issue (the agreement rate on *detection* was approximately 0.3), which is a unique characteristic of automated document revision that differs from traditional GECs.

5 A Case Study: (Meta) evaluation

In addition to creating a corpus for automated document revision, it is essential to establish an evaluation that can measure a document's quality improvement (and possibly deterioration) relative to the applied revisions. As a case study, we use TETRA to investigate reference-less and interpretable methods for a (meta-)evaluation method to detect quality improvements according to document revisions.

5.1 How do we evaluate revisions?

Ultimately, the evaluation of document revision systems itself is a research challenge that could be as difficult as building high-quality automated essay scoring (AES) systems (Dikli, 2006). A typical scenario for evaluating text generation is to compute the textual similarity between the hypothesis and references, as in machine translation (BLEU (Papineni et al., 2002)) and summarization

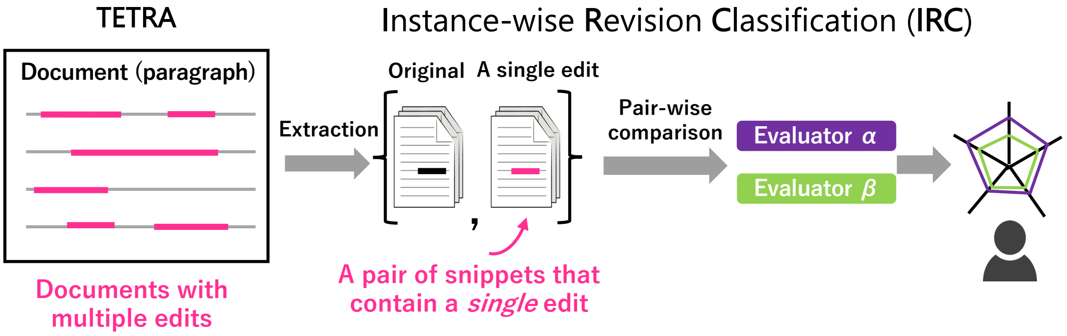

TETRA
Document (paragraph)
Documents with multiple edits
Extraction
Instance-wise Revision Classification (IRC)
Original
A single edit
A pair of snippets that contain a *single* edit
Pair-wise comparison
Evaluator α
Evaluator β

Figure 2: Overview of the IRC meta-evaluation with TETRA.

(ROUGE (Lin, 2004)). However, it is infeasible to elicit all possible gold references for document revision because there are infinite ways to edit a document. In fact, existing work using BLEU and ROUGE to evaluate document revisions shows that such reference-based metrics do not work due to the limited gold references (Du et al., 2022). In addition, given that the purpose of document revision is to support writing, simply presenting users (e.g., model developers and authors) with a single number (overall score) would be insufficient in terms of interpretability and transparency.

In light of the above, a good starting point for a first evaluation method for document revisions would be to develop an explanatory reference-free evaluation model for each evaluation perspective (e.g., clarity, readability, consistency) and then conduct a multidimensional evaluation using this model in an integrated manner.

5.2 Instance-wise revision classification

When using reference-free evaluation as described in §5.1, it is necessary to conduct a *meta-evaluation* of automatic evaluation models (evaluators) to see how well they correlate with human judgments and how reliable they are. Here, it is difficult to measure the quality of a revision automatically based on an *absolute* metric because a single document will contain a variety of edits based on many aspects of evaluation (Table 3). Thus, it is more straightforward to consider a *relative* metric, where a pair of documents is subject to a binary classification choosing the revised one. Such a pairwise comparison has been proven effective as a meta-evaluation method in cases where absolute evaluation is difficult (Guzmán et al., 2015; Christiano et al., 2017). Also, note that document revision contains multiple edits; thus, the binary prediction process cannot identify which edit(s) contributed to the improvement or the degree of improvement.

To address these concerns, we present **I**nstance-wise **r**evision **c**lassification (IRC) as a meta-evaluation methodology, where a pair of snippets that contain a *single* edit is given, and we compare the (reference-less) models according to the accuracy of the binary prediction (i.e., which of the snippets is a revision). By focusing on comparing 'single edit' differences, we can obtain transparent and interpretable measures for each type of edit (e.g., which edit type is more challenging to revise than other types). This is expected to enable us to investigate more effective evaluators (evaluation models) in the future. In fact, recent studies have demonstrated that such rubric-based interpretable evaluation correlates better with human judgments than single overall scoring techniques (Kasai et al., 2021a,b; Zhong et al., 2022). An overview of the proposed IRC is shown in Figure 2. The design philosophy of IRC is to provide users (e.g., model developers or writers) with analytical reports based on multidimensional evaluations to facilitate their understanding of the models, with the goal of moving away from chasing the highest overall number.

5.3 Experiment

In this subsection, we demonstrate how well existing large-scale pre-trained language models perform under the proposed IRC framework as (reference-less) models.

5.3.1 Data split

We divided TETRA into a training set (75%; 48 papers) and a test set (25%; 16 papers) to avoid paper overlap, and we converted the test data into pairs of snippets containing a single edit for IRC framework. Here, when multiple edit types were assigned, each edit type was extracted independently as a single edit snippet pair. When creating a pair of snippets,

	Grammaticality	Fluency	Clarity	Style	Readability	Redudancy	Consistency
BERT	**0.82**	**0.84**	**0.85**	**0.83**	**0.85**	0.79	**0.90**
GPT-4 zero-shot	0.42	0.57	0.55	0.59	0.46	0.47	0.58
+ explicit prompt	0.65	**0.79**	**0.67**	0.57	**0.71**	**0.92**	**0.62**
GPT-4 few-shot	0.47	0.48	0.56	0.51	0.44	0.45	0.40
+ explicit prompt	0.43	0.49	0.56	0.56	0.57	0.80	0.56

Table 6: Meta-evaluation result (Accuracy).

we extracted the entire paragraph as the context. In total, we extracted 1,368 snippet pairs for IRC meta-evaluation.

5.3.2 Evaluators

In this experiment, we compared BERT (Devlin et al., 2019) as fine-tuning and GPT-4 (OpenAI, 2023) as zero/few-shot settings to classify the original and single edit revision snippets.

BERT We converted the training set into a balanced positive/negative example by randomly swapping the order of snippet pairs in one-half of the training set. Specifically, we implemented this evaluator as a classification problem for the [CLS] tokens, using as input a sequence of tokens connecting the original and the single-edited revision documents with the [SEP] tokens. We used the PyTorch implementation for these Transformer models (Wolf et al., 2020). The hyperparameters used to train the model are shown in Appendix D

GPT-4 We build the model using the GPT-4 API (2024-02-15-preview) provided by OpenAI [7]. Two settings, zero-shot and few-shot (2-shot by following (Coyne et al., 2023)), were prepared to evaluate the performance with and without examples[8]. Furthermore, we created prompts focusing on text revision evaluation criteria (**explicit prompt**) to investigate the impact of prompts on evaluation performance, comparing them with the base prompt. Detailed information on each prompt is provided in Appendix E.

5.3.3 Results

As can be seen, the proposed IRC framework enabled us to evaluate the accuracy of each metric in terms of each aspect (i.e., edit type) while analyzing their strengths and weaknesses (Table 6). We also observe a significant disparity between fine-tuning and zero/few-shot results, highlighting the crucial role of fine-tuning in achieving automatic evaluation of text revision. Contrary to expectations, the LLM-based evaluator performed better in zero-shot compared to few-shot scenarios. One potential explanation is that presenting only a few cases might not only be insufficient but also noisy, especially in tasks involving diverse evaluation aspects and reasonable editing methods, such as text revision. On the other hand, compared to the base prompt, performance was significantly improved for many revision types when using explicit prompts. In particular, for redundancy, the GPT-4 evaluator with explicit prompt outperformed the finetuning model. This suggests the potential to realize an automatic evaluation model for high-performance text revision even for zero-shot by advancing prompt engineering in the future.

6 Analysis

The experimental results discussed in §5.3 demonstrated that the supervised metric can discriminate the original and revision snippets with reasonably high accuracy. However, the following question should be considered. *Is the high accuracy derived from actually detecting the quality improvement provided by the revision or annotation artifacts (spurious correlation) by commonly used words and phrases by expert annotators?*

To investigate this question, we evaluated the performance of *the same* supervised metric (BERT) used in §5.3 by applying corruption methods to TETRA in order to artificially degrade the quality of the source documents. If the same supervised metric is fine-tuned on the source and the (improved) revision can still select the original document over the degraded document, we can conclude that the metric actually distinguishes the *quality* of the document rather than spurious features.

6.1 Corruption Methods

Automatic Error Generation (AEG) Injecting grammatical errors as data augmentation has been studied actively to improve GEC. In this study, we

[7] https://github.com/openai/openai-python

[8] The example used for the few-shot was sampled from the train split.

used a back-translation model, which is the most commonly used model in GEC among AEG methods (Xie et al., 2018; Kiyono et al., 2019; Koyama et al., 2021), to deteriorate the original documents in terms of *grammaticality* and *fluency*. Here, a reverse model that generates an ungrammatical sentence from a given grammatical sentence was trained in the back-translation model. To construct the reverse model, we followed the general settings identified in previous studies (Kiyono et al., 2019; Koyama et al., 2021). The details of the experimental settings for the AEG model are described in Appendix F.

Sentence Shuffling As shown in Figure 1, the document revision process involves reordering sentences to improve the *flow* and *consistency* of argumentation. In this analytical experiment, after applying the AEG model, we further shuffled sentences with the same ratio as the *consistency* edit type (5% of the documents; refer to Table 3) to degrade the document relative to the sentence order.

6.2 Results

The binary classification accuracy obtained by BERT on the original vs. (degrading) corruption scenario was 0.96. We found that BERT can successfully select the original document over the degraded document. It should be noted that this is a simulation experiment with artificial errors and there are deviations from a realistic setting, but it suggests that the supervised baseline has the potential to learn to discriminate documents relative to quality rather than spurious features in the experts' annotations.

7 Conclusion

We have proposed the new document revision corpus and highlighted its uniqueness of it compared with existing corpora. As a case study using this corpus, we have explored reference-less and interpretable meta-evaluation methods and also demonstrated that a fine-tuned pre-trained language model can discriminate the quality of documents, which indicates the feasibility of automated document revision evaluation.

Limitations

The first limitation of this study is the scalability of the annotation. TETRA consists of *documents* revised by experts and is therefore expensive to scale up in its nature. This limitation could be mitigated by the choice of source data, i.e., there is room to replace experts with crowd workers by selecting source data that do not require expertise (e.g., general essays). We also reiterate that this work does not aim at proposing specific revision systems and evaluation models for automated document revision. Instead, we present a meta-evaluation scheme as a first step to develop such models and metrics with more transparency.

Ethics Statement

For developing a new document-level revision corpus, TETRA, we paid market rates to the professional editors for their annotations. With regard to the checklist items regarding the use and distribution of artifacts, none of the concerns apply to the dataset created in this study, as it was annotated based on the ACL Anthology materials. [9]

References

Linda Allal, Lucile Chanquoy, and Pierre Largy. 2004. *Revision Cognitive and Instructional Processes.*, volume 8. Springer.

Larry Beason. 1993. Feedback and revision in writing across the curriculum classes. *Research in the Teaching of English*, pages 395–422.

Eric Brill and Robert C. Moore. 2000. An improved error model for noisy channel spelling correction. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 286–293.

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 (ACL 2017)*, pages 793–805.

M. Buchman, R. Moore, L. Stern, and B. Feist. 2000. *Power Writing: Writing with Purpose.* No. 4. Pearson Education Canada.

Jill Burstein, Daniel Marcu, and Kevin Knight. 2003. Finding the write stuff: Automatic identification of discourse structure in student essays. *IEEE Intelligent Systems*, 18(1):32–39.

[9] https://aclanthology.org/faq/copyright/

Aoife Cahill, Nitin Madnani, Joel Tetreault, and Diane Napolitano. 2013. Robust systems for preposition error correction using Wikipedia revisions. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 507–517.

Hang Cao, Zhiquan Cao, Chi Hu, Baoyu Hou, Tong Xiao, and Jingbo Zhu. 2023. Improving autoregressive grammatical error correction with non-autoregressive models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12014–12027, Toronto, Canada. Association for Computational Linguistics.

Martin Chodorow, Joel Tetreault, and Na-Rae Han. 2007. Detection of grammatical errors involving prepositions. In *Proceedings of the Fourth ACL-SIGSEM Workshop on Prepositions*, pages 25–30.

Shamil Chollampatt and Hwee Tou Ng. 2018. A Multilayer Convolutional Encoder-Decoder Neural Network for Grammatical Error Correction. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI 2018)*, pages 5755–5762.

Leshem Choshen and Omri Abend. 2018. Reference-less measure of faithfulness for grammatical error correction. 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 124–129, New Orleans, Louisiana. Association for Computational Linguistics.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Steven Coyne, Keisuke Sakaguchi, Diana Galvan-Sosa, Michael Zock, and Kentaro Inui. 2023. Analyzing the performance of gpt-3.5 and gpt-4 in grammatical error correction.

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 (NAACL 2012)*, pages 568–572.

Robert Dale, Ilya Anisimoff, and George Narroway. 2012. Hoo 2012: A report on the preposition and determiner error correction shared task. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 54–62. Association for Computational Linguistics.

Robert Dale and Adam Kilgarriff. 2011. Helping our own: The hoo 2011 pilot shared task. In *Proceedings of the Generation Challenges Session at the 13th European Workshop on Natural Language Generation*, pages 242–249. Association for Computational Linguistics.

Mike D'Arcy, Alexis Ross, Erin Bransom, Bailey Kuehl, Jonathan Bragg, Tom Hope, and Doug Downey. 2023. Aries: A corpus of scientific paper edits made in response to peer reviews.

Vidas Daudaravicius, Rafael E. Banchs, Elena Volodina, and Courtney Napoles. 2016. A report on the automatic evaluation of scientific writing shared task. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 53–62. 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. Association for Computational Linguistics.

Semire Dikli. 2006. An Overview of Automated Scoring of Essays. *Journal of Technology, Learning, and Assessment*, 5(1).

Wanyu Du, Vipul Raheja, Dhruv Kumar, Zae Myung Kim, Melissa Lopez, and Dongyeop Kang. 2022. Understanding iterative revision from human-written text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3573–3590.

Lester Faigley and Stephen Witte. 1981. Analyzing revision. *College composition and communication*, 32(4):400–414.

Tao Fang, Jinpeng Hu, Derek F. Wong, Xiang Wan, Lidia S. Chao, and Tsung-Hui Chang. 2023. Improving grammatical error correction with multimodal feature integration. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9328–9344, Toronto, Canada. Association for Computational Linguistics.

Mariano Felice and Ted Briscoe. 2015. Towards a standard evaluation method for grammatical error detection and correction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NACL-HLT 2015)*, pages 578–587.

Rachele De Felice and Stephen G. Pulman. 2008. A Classifier-Based Approach to Preposition and Determiner Error Correction in L2 English. In *COLING*, pages 169–176.

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.

Linda Flower and John R Hayes. 1981. A cognitive process theory of writing. *College composition and communication*, 32(4):365–387.

Tao Ge, Furu Wei, and Ming Zhou. 2018. Reaching Human-level Performance in Automatic Grammatical Error Correction: An Empirical Study. *arXiv preprint arXiv:1807.01270*.

Peiyuan Gong, Xuebo Liu, Heyan Huang, and Min Zhang. 2022. Revisiting grammatical error correction evaluation and beyond. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6891–6902, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Takumi Gotou, Ryo Nagata, Masato Mita, and Kazuaki Hanawa. 2020. Taking the correction difficulty into account in grammatical error correction evaluation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2085–2095.

Francisco Guzmán, Shafiq Joty, Lluís Màrquez, and Preslav Nakov. 2015. Pairwise neural machine translation evaluation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 805–814. Association for Computational Linguistics.

Na-Rae Han, Martin Chodorow, and Claudia Leacock. 2006. Detecting Errors in English Article Usage by Non-Native Speakers. *Natural Language Engineering*, 12(2):115–129.

Kazuaki Hanawa, Ryo Nagata, and Kentaro Inui. 2021. Exploring methods for generating feedback comments for writing learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9719–9730, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Aminul Islam and Diana Inkpen. 2009. Real-word spelling correction using Google Web 1T 3-grams. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1241–1249.

Md Asadul Islam and Enrico Magnani. 2021. Is this the end of the gold standard? a straightforward reference-less grammatical error correction metric. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3009–3015, Online and Punta Cana, Dominican Republic. Association for 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 (ACL 2017)*, pages 753–762.

Chao Jiang, Wei Xu, and Samuel Stevens. 2022. arXivEdits: Understanding the human revision process in scientific writing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9420–9435, Abu Dhabi, United Arab Emirates. 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 (ACL 2020)*, pages 4248–4254.

Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Lavinia Dunagan, Jacob Morrison, Alexander R. Fabbri, Yejin Choi, and Noah A. Smith. 2021a. Bidimensional leaderboards: Generate and evaluate language hand in hand. *arXiv* `https://arxiv.org/abs/2112.04139`.

Jungo Kasai, Keisuke Sakaguchi, Lavinia Dunagan, Jacob Morrison, Ronan Le Bras, Yejin Choi, and Noah A. Smith. 2021b. Transparent human evaluation for image captioning. *arXiv* `https://arxiv.org/abs/2111.08940`.

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 1–35.

Diederik Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*.

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 2019)*, pages 1236–1242.

Charles W. Kneupper. 1978. Teaching argument: An introduction to the toulmin model. *College Composition and Communication*, 29(3):237–241.

Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 193–203, Tampere, Finland. European Association for Machine Translation.

Aomi Koyama, Kengo Hotate, Masahiro Kaneko, and Mamoru Komachi. 2021. 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.

Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel Tetreault. 2014. Automated grammatical error detection for language learners. *Synthesis lectures on human language technologies*, 7(1):1–170.

John Lee and Stephanie Seneff. 2008. Correcting misuse of verb forms. In *Proceedings of ACL-08: HLT*, pages 174–182.

John Lee and Jonathan Webster. 2012. A corpus of textual revisions in second language writing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 248–252, Jeju Island, Korea. Association for Computational Linguistics.

Yinghao Li, Xuebo Liu, Shuo Wang, Peiyuan Gong, Derek F. Wong, Yang Gao, Heyan Huang, and Min Zhang. 2023. TemplateGEC: Improving grammatical error correction with detection template. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6878–6892, Toronto, Canada. Association for Computational Linguistics.

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.

Yixin Liu, Alexander R. Fabbri, Jiawen Chen, Yilun Zhao, Simeng Han, Shafiq Joty, Pengfei Liu, Dragomir Radev, Chien-Sheng Wu, and Arman Cohan. 2023. Benchmarking generation and evaluation capabilities of large language models for instruction controllable summarization.

Masato Mita, Tomoya Mizumoto, Masahiro Kaneko, Ryo Nagata, and Kentaro Inui. 2019. Cross-corpora evaluation and analysis of grammatical error correction models — is single-corpus evaluation enough? 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 1309–1314, Minneapolis, Minnesota. Association for Computational Linguistics.

Ryo Nagata. 2019. Toward a task of feedback comment generation for writing learning. 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 3206–3215, Hong Kong, China. Association for Computational Linguistics.

Ryo Nagata, Masato Hagiwara, Kazuaki Hanawa, Masato Mita, Artem Chernodub, and Olena Nahorna. 2021. Shared task on feedback comment generation for language learners. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 320–324, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Ryo Nagata, Atsuo Kawai, Koichiro Morihiro, and Naoki Isu. 2006. A Feedback-Augmented Method for Detecting Errors in the Writing of Learners of English. In *COLING-ACL*, pages 241–248.

Ryo Nagata, Mikko Vilenius, and Edward Whittaker. 2014. Correcting preposition errors in learner English using error case frames and feedback messages. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 754–764.

Courtney Napoles, Maria Nădejde, and Joel Tetreault. 2019. Enabling robust grammatical error correction in new domains: Data sets, metrics, and analyses. *Transactions of the Association for Computational Linguistics*, pages 551–566.

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 (ACL-IJCNLP 2015)*, pages 588–593.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2016. There's no comparison: Referenceless evaluation metrics in grammatical error correction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2109–2115. 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 (EACL 2017)*, pages 229–234.

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 18th Conference on Computational Natural Language Learning (CoNLL 2014): Shared Task*, pages 1–14.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 Shared Task on Grammatical Error Correction. In *Proceedings of the 17th Conference on Computational Natural Language Learning (CoNLL 2013): Shared Task*, pages 1–12.

OpenAI. 2023. Gpt-4 technical report.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2019)*.

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.

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.

Alla Rozovskaya and Dan Roth. 2014. Building a state-of-the-art grammatical error correction system. *Transactions of the Association for Computational Linguistics*, 2:419–434.

Keisuke Sakaguchi, Courtney Napoles, Matt Post, and Joel Tetreault. 2016. Reassessing the goals of grammatical error correction: Fluency instead of grammaticality. *Transactions of the Association for Computational Linguistics*, 4:169–182.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 1715–1725.

Anthony Seow. 2002. *The Writing Process and Process Writing*, page 315–320. Cambridge University 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 (ACL 2012)*, pages 198–202.

Joel Tetreault, Jennifer Foster, and Martin Chodorow. 2010. Using parse features for preposition selection and error detection. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 353–358.

Joel R. Tetreault and Martin Chodorow. 2008. The ups and downs of preposition error detection in ESL writing. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 865–872.

Kristina Toutanova and Robert Moore. 2002. Pronunciation modeling for improved spelling correction. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 144–151.

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 31 (NIPS 2017)*, pages 5998–6008.

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. Association for Computational Linguistics.

Ziang Xie, Guillaume Genthial, Andrew Y. Ng, and Dan Jurafsky. 2018. Noising and Denoising Natural Language: Diverse Backtranslation for Grammar Correction. In *NAACL*, pages 619–628.

Lingyu Yang, Hongjia Li, Lei Li, Chengyin Xu, Shutao Xia, and Chun Yuan. 2023. LET: Leveraging error type information for grammatical error correction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5986–5998, Toronto, Canada. Association for Computational Linguistics.

Jingheng Ye, Yinghui Li, Qingyu Zhou, Yangning Li, Shirong Ma, Hai-Tao Zheng, and Ying Shen. 2023. CLEME: Debiasing multi-reference evaluation for grammatical error correction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6174–6189, Singapore. 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.

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. Association for Computational Linguistics.

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, San Diego, California. Association for Computational Linguistics.

Fan Zhang and Diane Litman. 2014. Sentence-level rewriting detection. In *Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 149–154, Baltimore, Maryland. Association for Computational Linguistics.

Fan Zhang and Diane Litman. 2015. Annotation and classification of argumentative writing revisions. In

Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 133–143, Denver, Colorado. Association for Computational Linguistics.

Yue Zhang, Leyang Cui, Enbo Zhao, Wei Bi, and Shuming Shi. 2023. RobustGEC: Robust grammatical error correction against subtle context perturbation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16780–16793, Singapore. Association for Computational Linguistics.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation.

A Recruitment procedure for annotators

We recruited professional editors who are native speakers of English and have domain expertise in academic writing, directly via Upwork (`https://www.upwork.com/`), a freelance marketplace, through interviews and screening tests to ensure the quality of the annotators. We paid market rates to them. Instead of using the services of an English proofreading company, which tends to be uncontrollable in terms of annotator quality, we directly hired annotators and provided them with feedback to control the annotation quality, which contributed to further improving the dataset's quality. We will extend the description of this annotation process in the camera ready.

B Instructions for annotators

The full text of the instructions to the annotators is reported below.

Summary You will be proofreading and editing the abstracts and the introduction sections of scientific papers published at NLP (Natural Language Processing) conferences and workshops. Please make edits to improve the quality of the papers, along with your comments mentioning what aspect of the paper the edit is intended to improve, without changing the meaning of the content (information contained in the paper).

About the papers

- These papers are randomly chosen from a pool of papers published at recent NLP conferences and workshops.
- These papers are written by a diverse set of authors, including native and non-native speakers of English at various stages of their careers (students, researchers, faculty members, etc.).
- These papers went through peer reviews and were accepted at conferences and workshops

Edits

- Make edits to the papers in order to improve their quality without changing the information contained in the papers. For each edit, mention what aspect of the paper the edit is intended to improve. These aspects include, but are not limited to: Mechanics, punctuation, grammar, spelling, word order, word usage, organization, development, cohesiveness, coherence, clarity, content, consistency, voice. Feel free to use your own tags/words to describe the purpose of your edit
- Refrain from making single edits that improve more than one aspect of the paper at the same time. Make two or more separate, overlapping edits in the same place if you need to improve multiple aspects.
- Feel free to be creative and make changes that span over multiple sentences or ones that rearrange sentences or even paragraphs if necessary. You are encouraged to rewrite the sentences and paragraphs if local edits aren't enough to improve the quality.
- Since these papers are already peer-reviewed, we expect fewer low-level edits related to punctuation, spelling, and grammar, although make sure to correct such errors if you do encounter them.
- Focus instead on types of edits that improve higher-level aspects of the paper (such as organization, development, cohesiveness, coherence, clarity, content, voice, etc.)

C Example of XML annotation

See Table 7.

D Hyper-parameters settings

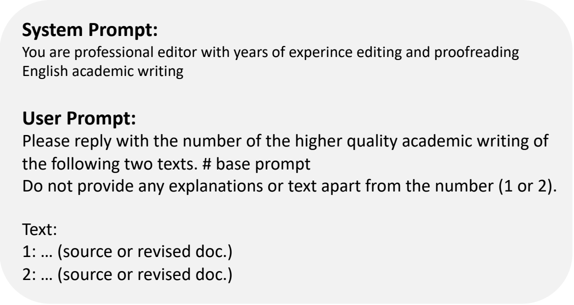
See Table 8.

E Prompts in the GPT-4 evaluators

The prompt used for GPT-4 evaluator is illustrated in Table 3. For prompts focused on evaluation criteria, the following sentence was replaced with base prompt.

```
<doc id="Pxx-xxxx" editor="A" format="Conference" position="Non-student" region="Native">
<abstract>
<text>In this paper, (...) extracted sense inventory. The</text>
<edit type="conciseness" crr="induction and disambiguation steps" comments="conciseness - just
    tightening it up a little bit.">induction step and the disambiguation step</edit>
<text>are based on the same principle: (...) topical dimensions</text>
<edit type="readability" crr=". In" comments="readability - this sentence is getting a bit long, so
    splitting it in two here.">; in</edit>
<text>a similar vein, ...</text>
...
</abstract>
<introduction>
<text>Word sense induction (...)</text>
<text>\n\n Word sense disambiguation (...)</text>
<edit type="punctuation" crr="" comments="punctuation - comma is not appropriate.">,</edit>
...
</introduction>
```

Table 7: Example of XML annotation. For brevity, we omitted a part of the text with "...".



System Prompt:
You are professional editor with years of experince editing and proofreading English academic writing

User Prompt:
Please reply with the number of the higher quality academic writing of the following two texts. # base prompt
Do not provide any explanations or text apart from the number (1 or 2).

Text:
1: ... (source or revised doc.)
2: ... (source or revised doc.)

Figure 3: Example of prompt.

Configurations	Values
Model Architecture	bert-base-uncased
Optimizer	Adam (Kingma and Ba, 2015)
Learning Rate	2e-5
Number of Epochs	10
Batch Size	32

Table 8: Hyper-parameters settings

- Grammaticality: "Please reply with a more grammatical text number."
- Fluency: "Please reply with a more fluent text number."
- Clarity: "Please reply with the number of the text whose meaning is clearer."
- Style: "Please reply with the number of the higher quality academic writing of the following two texts. Please focus your evaluation on the adaptation to an academic writing style in particular."
- Readability: "Please reply with a more readable text number."
- Redundancy: "Please reply with a text number that is less redundant."
- Consistency: "Please reply with more consistent text."

F Experimental settings for AEG

We adopted the "Transformer (big)" settings (Vaswani et al., 2017) using the implementation in the `fairseq` toolkit (Ott et al., 2019) as a GEC model. In addition, we used the BEA-2019 workshop official dataset (Bryant et al., 2019) as the training and validation data. For preprocessing, we tokenized the training data using the `spaCy` tokenizer[10]. Then, we removed sentence pairs where both sentences where identical or both longer than 80 tokens. Finally, we acquired subwords from the target sentence via the byte-pair-encoding (BPE) (Sennrich et al., 2016) algorithm. We used the `subword-nmt` implementation[11] and then applied BPE to split both source and target texts. The number of merge operations was set to 8,000.

[10] https://spacy.io/

[11] https://github.com/rsennrich/subword-nmt

Evaluating Vocabulary Usage in LLMs

Matthew Durward and **Christopher Thomson**
University of Canterbury
matthew.durward@pg.canterbury.ac.nz
christopher.thomson@canterbury.ac.nz

Abstract

In the rapidly evolving educational technology landscape, the potential applications and limitations of AI-generated content need greater scrutiny. This study explores the authenticity of AI-generated texts by comparing vocabulary usage between human-authored texts and those generated by AI across different registers, specifically in news and creative writing. Employing Vocabulary-Management Profiles (VMPs) for structural analysis and word keyness analysis to evaluate vocabulary frequency and dispersion, we reveal distinct patterns of text production. Our results demonstrate variation in vocabulary usage between human and AI-generated texts across registers, and shows how VMPs capture these differences effectively. These findings highlight the challenges Large Language Models (LLMs) face in mimicking human text generation and open some new avenues for examining characteristics of vocabulary use relevant to applications in education.

1 Introduction

We are navigating a transformative era, marked significantly by the integration of AI technologies into various aspects of daily life. This is particularly evident in the realm of language learning, where Large Language Models (LLMs) have become instrumental. LLMs find application across diverse sectors including education, healthcare, and research, showcasing their versatility and impact (Hosseini et al., 2023). The role of LLMs in language acquisition and written composition deserves special attention; it is claimed they offer substantial benefits to learners through personalized learning experiences, interactive prompts for questions and examples, and feedback on writing (Dao, 2023). This highlights the potential of LLMs to enhance the efficacy of language learning strategies significantly.

While the potential is certainly undeniable, a factor that is worth addressing is whether texts produced by LLMs - particularly in the form of examples generated in a learning environment - accurately represent what a learner is likely to observe in a real-world scenario. In particular, we aim to gain a better understanding of whether LLMs generate text with respect to different registers in a fashion similar to humans.

Previous research (AlAfnan and MohdZuki, 2023; Gómez-Rodríguez and Williams, 2023) provides some insight into the perceived 'style' and characteristics of LLM production. We narrow our scope to focus on attributes related to discourse and vocabulary, two adjacent concepts that we expect to differ by register. In particular, we are interested in how vocabulary is deployed within the structure of texts. Anecdotally, LLM text is often described as 'generic' or 'bland' in tone. Thus, we were motivated to understand the extent to which such differences are linked to lexical diversity and the rate, or sequencing, with which new vocabulary is introduced. To achieve this, we investigate human authored and machine generated texts through Vocabulary-Management Profiles (VMP). In their simplest interpretation, VMPs provide a linear representation, that can be graphically illustrated, representing the rate of newly introduced vocabulary through the progression of a text.

2 Related Work

2.1 AI text for Language Acquisition and Development

Language learners and teachers are enthusiastic about LLMs, but research on their pedagogical uses is still in infancy. Researchers Kostka and Toncelli (2023) highlight the opportunities of these systems in an English Language Learning setting and advocate for cross-collaboration between educators, students, and developers. We are at a point where LLM systems are being adopted in an ever-growing manner, and efforts are needed to

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 266–282
June 20, 2024 ©2024 Association for Computational Linguistics

understand what differentiates AI-generated text from authentic human-authored text and what consequences may flow from these differences in an educational setting. For instance, Vaccino-Salvadore (2023) outline areas of concern and ethical considerations when bringing systems like ChatGPT into the classroom for language learning, especially the bias and diversity constraints inherent in these systems. We must remember that systems like ChatGPT and similar LLMs derive their training data largely from the internet, potentially reproducing or amplifying cultural and linguistic biases, replicating dominant themes and linguistic patterns, and reducing the diversity of language compared to what is actually in use around the world (Ray, 2023).

Bringing LLMs into a language learning setting involves many considerations. While research (Baskara and Mukarto, 2023) highlights the potential benefits, such as the personalization, or generation, of authentic learning materials, more work on how these systems differ from human-generated text would be beneficial. We also note that recent work on the degradation of LLMs trained on synthetic data such as (Shumailov et al., 2023) and (Guo et al., 2023) suggests that small reductions in quality or diversity of learning materials can, if propagated, be catastrophic for language models. We must understand how text generated by LLMs differs from human-authored text to evaluate these synthetic materials properly for human language instruction.

2.2 Vocabulary Management Profiles

Vocabulary-management profiles provide a method for measuring the rate at which new vocabulary is introduced throughout a text and a convenient means of representing this graphically (Youmans, 1991, 1994). Prior to developing VMPs, Youmans's (1990) worked on conceptually similar graphic representations through the broader application of type-token vocabulary curves (TTVC), their derivations, and their estimation of vocabulary size. Type-token modelling- examining the ratio of the number of unique word (types) and number of collective words (tokens), are readily examined in the field of linguistics (Mitchell, 2015) and the area of language development and acquisition (Jarvis, 2013). Token curves approximate lexical diversity (LD) progression over time (or length of a text), but compared to VMPs, they offer a more generalized indication of lexical usage within a text. On observing a TTVCs, we can relate curves with a more pronounced increase in lexical diversity, whereas shallower curves denote a lower lexical diversity. VMPs improve on these earlier methods by observing the number of new (word-) types that occur in a moving window across the text. The difference is that VMPs aim to move beyond a static lexical assessment of the text as a whole and instead observe local patterns in the sequence of vocabulary use. Relating these structures to narrative or text structure, different slope trajectories can indicate factors signalling boundary points or "a new turn in the story" (Stubbs, 2006, p. 142). These turns can reveal the author's stylistic attributes and narrative structure. Indeed, they display a necessary storytelling component, balancing the inclusion of new words to help progress a story with the repetition of older words that help ensure text cohesion (Stubbs, 2006). Close analysis of these structural changes can expose how a writer navigates changes in topic and style or how diegesis relates to exegesis (Clement, 2013).

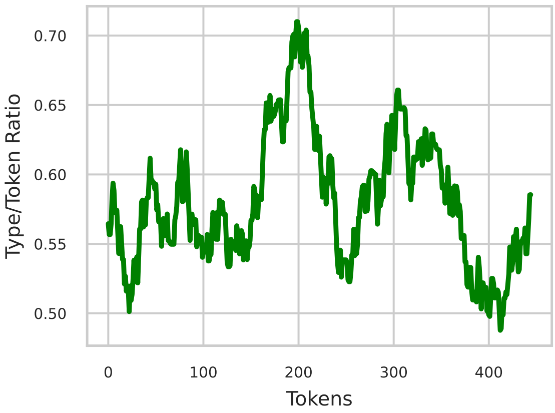


Figure 1: VMP Curve for a human-authored creative writing sample. The Type/Token Ratio (y-axis) averages individual word ratios of a moving window (set at 51 here) across a text, plotted against the token sequence (x-axis) of the text sample

The advantage of investigating text in this manner is that it provides a structure resembling that found more traditionally in time-series data, which enables a flexible perspective of the scope of a text, peering into not only global trends but also narrowing the field into patterns that when they emerge can provide insight into representations of different groups of text. By observing an individual text under the lens of a VMP, relationships emerge as to the dynamics indicated through respective peaks and valleys "signaling the ebb and flow" of information in texts (Youmans, 1991, p. 4). Youmans suggests that vocabulary used less frequently towards the end of moving intervals is often associated with introducing new topics. In contrast, vocabulary used more recently is likely to indicate the continuation of an existing topic. Evans (2020) refers to this as fractal patterning and attributes these formulations as evidence for the nested dynamics of self-similar attributes found from a global reference such as a novel with self-reciprocating emerging patterns of peaks and valleys, demarcated between narrowing orders of magnitude, as in sections, chapters, and paragraphs.

The degree of effectiveness of VMPs will vary with different applications. McKenny (2003) applied VMPs to ELL essays and observed their capacity to identify texts that follow stylistic choices of including new information as an inspirational factor for concluding remarks. However, because VMPs generalise patterns in the introduction or regularity of vocabulary use, they cannot substitute for context-specific qualitative analysis. For example, Meyer and Cooney (1994) found that VMPs benefit textual analysis by providing insight into the use of new and known information as measured by vocabulary but express limitations, particularly in the case of contextual usage, or *how* a word is used. This acknowledgement aligns with McKenny's (2003) positing the need for clear objectives when generalising about VMPs.

2.3 Word Keyness and Dispersion

In an educational context, the concept of 'Keyness' is closely aligned with creating word lists for language learning, emphasizing the strategic selection of high-frequency vocabulary. Nation (2006) highlights the value of these lists in planning vocabulary learning, a notion supported by further research (Nation, 2011). Vocabulary selection, tailored to learners' needs and specific domains like academia, plays a pivotal role. Such specialized lists, as Nation (2006) notes, are highly generalizable, proving effective across disciplines and extending to journalistic language. This effectiveness is further confirmed in English for Specific Purposes (ESP), showcasing the utility of targeted vocabulary strategies (Đurović, 2023).

To elucidate the variation in lexical usage across our corpora from a broad perspective, we implement two methods to measure a word's *keyness*. First, concerning word frequency, then second, we employ dispersion measures to discern between AI-generated and human-generated text. Dispersion assesses how evenly or unevenly a word is distributed within a corpus. Aiming to identify keywords that differentiate humans from AI-generated text, relying solely on frequency lists may fall short of offering a comprehensive understanding of vocabulary usage. Dispersion offers more profound insights into lexical patterns, as our analysis spans diverse sources (i.e., human and AI) and various genres (news versus creative writing). Given prior studies on source and genre variation (Biber, 1987; Kruger and Rooy, 2018), dispersion effectively provides a holistic view of vocabulary disparities. In contrast, the VMP analysis yields a more detailed, text-specific exploration of these differences.

Keyness broadly reflects a word's presence and significance in a corpus relative to its size, highlighting the word's distribution and importance (Jeaco, 2023). It is closely linked with dispersion, helping identify core vocabulary differences between corpora. Building on Egbert and Biber (2019)'s work on incorporating dispersion in keyword analysis, we apply Gries's (2021) method for a nuanced assessment of keyness. This method evaluates a word's frequency and dispersion to determine its unique role across corpora more accurately, avoiding biases introduced by frequency-based measures, such as the log-likelihood ratio. This approach enables a detailed comparison of vocabularies, offering insights into distinctive lexical patterns (Gries, 2021).

Our research aligns with the goals of authentic material matching used in a language learning context. Briefly, while there are competing notions, authenticity is described here as genuine language used in writing to communicate a meaningful message to a real audience, encompassing a wide variety of language (Gilmore, 2007; Morrow, 1977). There are numerous ways of measuring aspects of authenticity concerning discourse and lexical diversity, such as register variation multi-dimensional analysis (Biber, 2014) or linguistic feature extraction (Lee and Lee, 2023). By restricting our focus to vocabulary, we can disseminate variation in a manner easily processable by educators and learners. Often, overly complex systems with a multitude of features can add dimensions of entanglement, making it difficult for users to interpret results. VMPs are positioned to provide graphical representations that provide indications of the rate of introduced vocabulary where patterns are visually identifiable and computationally measurable. Through VMP and Keyness analysis, we can extract vocabulary information that can be conveyed intuitively, making identifying patterns readily understood by a spectrum of users.

3 Method

- RQ1: Are there distinguishable patterns of vocabulary usage across different text sources and registers?
- RQ2: In what ways does analyzing texts through VMPs uncover structural differences across sources and registers?
- RQ3: How do frequency and dispersion-based keyness analyses reveal vocabulary patterns across various text sources and registers?

We aim to investigate vocabulary usage through two distinct lenses. By implementing Vocabulary-Management Profiles (VMPs), we evaluate structural differences in writing patterns, shedding light on how texts from various sources and registers unfold. Then, through *word* keyness analysis, we qualitatively examine words associated with specific sources and registers to grasp salient differences through vocabulary usage better.

3.1 Data

This study investigates text under two dimensions of consideration: the source of the data (i.e. human or AI) and the register (i.e. news or creative). Data was retrieved from the DeepfakeTextDetect [1] dataset. Further details regarding the compilation of the initial dataset can be found in (Li et al., 2023). This combined dataset comprises eight different registers and text generated from 27 LLMs. We create a subset of extracted text from LLM sources, OpenAI gpt-turbo-3.5 and Meta LLaMA 65B, along with their human-generated counterparts. The length of a text plays a role in the observations of VMPs. Simply put, the longer a text is, the more observations can be extracted. Unfortunately, LLM prompts often generate texts well below what humans produce. We selected texts within a range of 400 to 500 tokens to strike a balance. Token counts are obtained after a preprocessing stage where words are converted to lowercase and punctuation is removed. Additionally, Youmans's (1991) found that further preprocessing modifications, such as removing affixes and conflating synonyms, have a minimal impact on the graphical representations of English discourses, so we refrained from any lemmatization or stemming procedures. This yielded 120 texts for each unique register/source combination, totalling 720 individual texts for analysis. Details of the corpus can be viewed in Table 1.

Source	Mean	Standard Deviation
creative_65B	448.70	29.62
creative_gpt	430.95	23.66
creative_human	450.47	28.75
news_65B	450.52	30.29
news_gpt	427.64	25.13
news_human	448.75	28.29

Table 1: Mean and Standard Deviation of Word Counts (in tokens) by Source.

3.2 Vocabulary Management Profiles

As discussed, VMPs can be thought of as a moving window through the progression of a text that measures the rate of newly introduced vocabulary. Youmans developed three methods for calculating VMPs (Youmans, "How to generate VMP 2.2s"); we use the VMP 2.2 method here. Our `vmp` function takes three parameters: `delta_x`, which is the size of the moving window; `cleaned_tokens`, a list of preprocessed tokens from the text; and for convenience, we specify `half_delta_x`, the middle value of the moving window, to be used for plotting. The function operates by sliding a window across the `cleaned_tokens`, giving each new vocabulary word a score of 1.0, and for each repeated word, determining a score using the following calculation: "(Number (index) of Current Word - Number (index) of Previous Occurrence - 1)/(Total tokens in the Text - 1)" (adapted from Youmans, "How to generate VMP 2.2s"). To ensure scores for the start of the text are consistent, the moving window centred on token 1 of the text 'wraps around' so that its first half covers the end of the text. This way, the VMP 2.2 measures vocabulary use at a 'second pass' through the text. (Youmans "How to generate VMP 2.2s").

Some considerations worth noting are the user-defined parameters. First, how a user wishes to treat common words and other preprocessing. We are interested in VMPs as a potential measure of stylistic and structural/topical changes, so we present results with common words retained (commonYes) and without (commonNo). Beyond this we have set aside investigation of the effects of different

[1] https://huggingface.co/datasets/yaful/DeepfakeTextDetect

kinds of preprocessing in the current study. An important parameter is the `delta_x` value. This value corresponds with the window size moving over each text. While Youmans (1991) suggests that longer `delta_x` values would be better suited for long-term patterns, it is also observed as having a smoothing effect on the trend of `delta_y` through a text. We examine a range of window sizes and suggest some additional smoothing techniques. A package to generate VMPs can be found at `github.com/matthewdurward/vmp`.

3.3 Keyness: Frequency and Dispersion

We investigate two independent properties related to vocabulary, *keyness* as it relates to frequency and again as it relates to dispersion. The combining of both is what Egbert and Biber (2019) describe as *key* keywords or words that demonstrate the collective power of both elements. To calculate keyness concerning frequency, we apply Gries's (2021) adaptation of Kullback-Leibler (KL) divergence to capture a word's association with a corpus. Equation (1) presents a generalized form of the Kullback-Leibler divergence, D_{KL} used to evaluate the extent of divergence between the conditional probabilities by observing a specific *word* in two corpora, compared to the overall probabilities within those corpora.

Equation (2) provides the calculation in application that measures how one probability distribution diverges from a second, expected probability distribution. In the context of text analysis, D_{KL} can be used to compare the distribution of word frequencies in one corpus or document (the "target") against another (the "reference"). A higher value of D_{KL} indicates a more significant divergence between the two distributions. If the divergence is zero, the two distributions are identical.

$$D_{KL}(p(\text{corpus} \mid \textit{word}) \,||\, p(\text{corpus})) \tag{1}$$

$$\left(a \times \log_2 \frac{a}{e}\right) + \left(b \times \log_2 \frac{b}{f}\right) \tag{2}$$

$$a = \frac{\text{Occ. of } \textit{word} \text{ in Target corpus}}{\text{Total Occ. of } \textit{word} \text{ in Target + Reference}} \tag{3}$$

$$b = \frac{\text{Occ. of } \textit{word} \text{ in Reference corpus}}{\text{Total Occ. of } \textit{word} \text{ in Target + Reference}} \tag{4}$$

In Equation (2), a signifies the relative frequency of a specific word in our target corpus (e.g., human news) compared to its presence in both the target and reference corpora (e.g., human news + GPT news) as illustrated in (3). Conversely, b indicates this word's relative frequency within the reference corpus (e.g., GPT news), also in relation to the combined target and reference, shown in (4). The variables e and f represent the proportion of all words in the target and reference corpora, respectively, to the total word count across both. The sign, or direction, of D_{KL} for frequency remains positive when the *word* in question prefers the Target corpus ($a > b$) and set to negative when the *word* prefers the Reference corpus ($b > a$). Thus, D_{KL} for frequency provides two aspects of consideration, the magnitude or strength of divergence and the direction of favorability for a corpus. Essentially, equation (2) quantifies how the distribution of a particular word differs between two textual datasets, helping to ascertain its distinctiveness or prevalence within one corpus as opposed to the other.

To compute dispersion, we adopt the methodology outlined by (Gries, 2021), utilizing the D_{KL} calculation previously employed to assess keywords for frequency. This method now serves as an analytical tool to gauge the distribution of a word's occurrence across different corpus segments, contrasting its distribution in one part of the corpus (target or reference) with the other parts. Applying this information-theoretic metric allows us to evaluate the frequency and spread of lexical items, providing nuanced insights into their usage patterns within and across corpora.

A normalization step is applied, $1 - e^{-D_{KL}}$, to transform the Kullback-Leibler divergence, D_{KL}, which can potentially range from 0 to ∞, into a value that falls within the closed interval [0, 1]. This transformation ensures that the dispersion measure is bounded and interpretable. Lower values of the normalized dispersion indicate less divergence from the expected distribution, whereas values closer to 1 suggest greater divergence.

3.4 Transformation and Dynamic-Time Warping

We revisit VMPs as a method for textual analysis, treating texts as time-series data to explore their dynamics using time-series analysis methodologies. To understand the stylistic and lexical variations across different sources and registers, we applied

Dynamic Time Warping (DTW). DTW functions as a measure of distance between two distinct text VMPs, with the progression of words representing time, and the type/token ratio of the VMPs serving as the unit of measurement. Recognizing the challenge posed by the variability and noise in raw VMPs, we preprocessed the data with wavelet denoising and Gaussian smoothing. This approach, employing the 'db1' wavelet for denoising and a sigma of 2 for smoothing, effectively minimized noise and highlighted long-term trends without sacrificing the VMPs' core characteristics.

These preprocessing steps clarified the VMPs for better interpretability and enhanced their analysis with DTW, allowing us to identify both subtle and pronounced differences in vocabulary usage. This nuanced examination, facilitated by signal processing techniques, affirms VMPs' utility for our context. Figure 2 illustrates an example of the comparative analysis of two separate pairs of individual text VMPs marked as similar (A) and dissimilar (B) using DTW. We calculated pairwise DTW distances of extracted VMPs both within the same register/source (such as creative/human) and between different sources but the same register (for example, news/65B compared to news/human). From these calculations, we derived distance matrices that were transformed into self-similarity measures. These measures are scaled between 0 and 1, where values closer to 1 indicate a higher similarity between a specific pair of Vocabulary Management Profiles (VMPs).

3.5 VMP Characteristic Features

Our analysis covers the interaction between different registers and sources, examining various conditions such as window sizes and the inclusion of common words. In this context, DTW serves as a method to quantify structural similarities. Beyond DTW, we further investigate time-series characteristics of the VMP themselves. To quantitatively assess the observed disparities, we employed three specific time series characteristic features: `DN_mean`, `DN_Spread_Std`, and `MD_hrv_classic_pnn40`. These features were derived using the *catch22* package, details of which can be found in the repository[2].

`DN_mean` computes the average Type/Token ratio across the series, serving as a measure of lexical diversity within the text. Higher values indicate

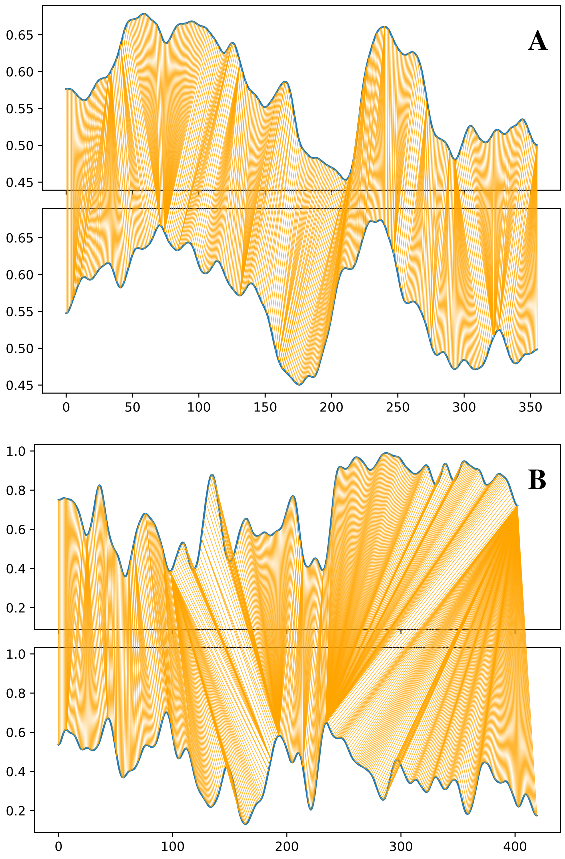


Figure 2: Dynamic Time Warping (DTW) visualizations illustrating the variability in VMP profiles for two pairs of example texts. Image (A) depicts a warping path with the minimal DTW distances, which suggests a closer similarity between 'creative-human' and 'creative-gpt' sample text sequences using a window of size 51. Image (B) presents a warping path with maximal DTW distance, where the orange lines exhibit more deviations, indicating substantial differences. This example uses a window size of 11. in the temporal patterns of 'news-human' and 'creative-65B' sample text sequences over a window of size 11. These paths reflect the level of adaptation required to align the sequences, with a more veritcal path implying less adjustment and a deviated path indicating more significant temporal distortion.

a greater variety of words used. `DN_Spread_Std` measures the spread of the Type/Token ratios around the mean, quantifying the variability in lexical diversity across different text segments. Lastly, `MD_hrv_classic_pnn40` denotes the proportion of significant incremental changes within the series, effectively capturing the frequency and magnitude of fluctuations in lexical diversity. A higher value suggests more pronounced and rapid shifts in the Type/Token ratios, reflecting erratic changes in the VMP. Further details of features and extraction methods are described in (Lubba et al., 2019).

[2] https://github.com/DynamicsAndNeuralSystems/catch22

4 Results and Discussion

Initial observations comparing VMPs, as illustrated in Figure 3, reveal notable differences across all conditions of varying window sizes for both registers, represented in scenarios of excluding common words (Figure 6) and including common words (Figure 7).

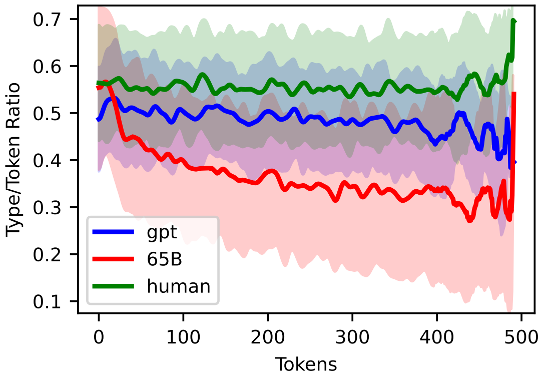


Figure 3: Sample VMP plot for the creative register texts by source including common function words with a window size of 11. It displays mean lines for the source groups, with variability indicated by shaded areas representing one standard deviation from the mean.

4.1 VMP Characteristics

To answer RQ1 and RQ2, statistical examination across various conditions, including common words and window size, has revealed significant distinctions among group sources for each of our tested VMP characteristic features. Our findings reported in this section exhibit highly significant p-values (p < 0.00001) for Kruskal-Wallis tests. Therefore, we emphasize the results with the most robust effect size, eta squared. From a broad perspective, human writing can be generalized as having higher lexical diversity represented through higher DN_Mean scores and higher consistent variability as demonstrated in DN_Spread_Std. Conversely, 65B demonstrates more sporadic episodes in texts with a generally higher MD_hrv_classic_pnn40. Notably, the most significant effect sizes were predominantly found in the news register, particularly for a window size of 25. For the feature DN_Mean in the commonNo category, a significant effect size of 0.4394 underlines a marked distinction primarily between 65B and human-generated texts, as well as between human and gpt variants. This difference points to the human-generated texts generally having higher Type/Token ratios than their counterparts.

Analyzing the DN_Spread_Std feature within the context of news content, particularly for the Delta 9, commonNo condition, provides insight into the variability of textual production across different sources. The effect size of 0.1955 indicates substantial variability differences among the groups, particularly between GPT and human VMPs. Posthoc comparisons further elucidate the nature of these differences: while both comparisons involving the 65B model (against GPT and human) showed significant results, indicating 65B's distinct variability profile, the direct comparison between news_gpt and news_human did not reach statistical significance (p=0.3882). The MD_hrv_classic_pnn40 feature further highlighted significant disparities, most notably in the news content for Delta 35, commonNo, with an effect size of 0.0815, particularly evident in comparing news for 65B and GPT.

4.2 DTW Based Similarity for VMPs

To provide a broader perspective on the variations in distributions of VMPs, we transformed DTW distance scores between pairs of VMPs into self-similarity scores. This approach facilitates a comparative analysis of textual characteristics across different registers and sources, visualized in Figure 4 and further detailed by condition in Figure 8. Our analysis reveals that human-generated texts, particularly in the news register without common words and with a window size of 25, consistently demonstrate the highest values for our tested features, underscoring the distinctiveness of human linguistic patterns compared to those generated by AI models such as GPT and 65B.

To assess the statistical significance of observed differences between the creative and news registers within each source, we conducted Mann-Whitney U tests. Given the multiple comparisons made, we applied the Bonferroni correction. Our results showed highly significant differences between registers for all sources, with all adjusted $p < 0.00001$, demonstrating robust disparities. While it was expected that there would be differences between registers for our source, our attention relates to the effect size of our comparisons.

The effect size for these differences was quantified using the rank biserial correlation, which emphasizes the direction and magnitude of disparity between registers of the same source. This ap-

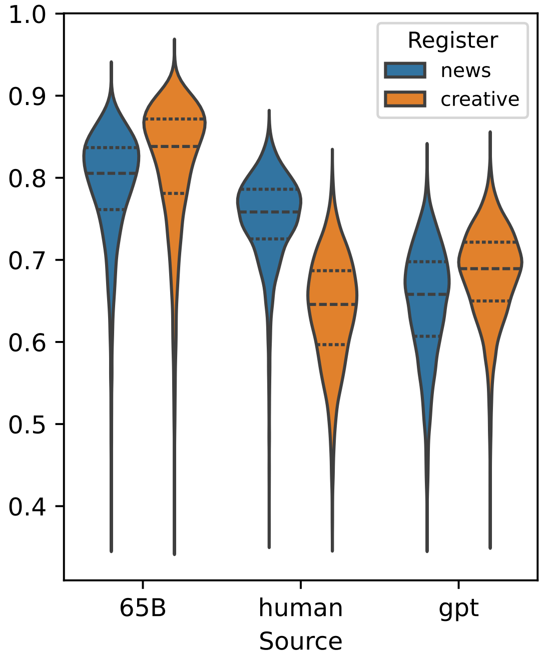


Figure 4: Distribution of DTW converted self similarity scores by source type for VMPs with a window size of 11, using all vocabulary. The violin plots illustrate score distributions across creative and news registers for 65B, human, and GPT sources, emphasizing the variations within each source and the distinctions between registers. Higher values indicate greater self similarity.

proach highlights each register and source's distinct linguistic features and VMP characteristics. As we can see in Figure 4, which shows results for the condition of a window size of 11 and no filtering of common words (CommonNo), we note that there are noticeable differences between news and creative VMPs for our sources.

A noteworthy observation pertains to the self-similarity within registers for each source group. Specifically, the human source group exhibits greater self-similarity and a more concentrated distribution for the news register, in contrast to a broader distribution and lower self-similarity for the creative register. Conversely, the 65B and GPT sources exemplify an opposite trend, with variations in self-similarity and distribution patterns. Table 2 indicates the most pronounced disparities are observable within the human source category, which consistently demonstrates the most substantial effect sizes, as denoted by r, thereby indicating a distinct separation between the news and creative registers. This distinction underscores the efficacy of text VMPs in differentiating between registers. Another focal point is the directional tendency of the correlation.

Source	r (CommonYes)	r (CommonNo)
	Interval: 9	
65B	-0.076	-0.317
Human	**0.945**	**0.654**
GPT	0.038	-0.410
	Interval: 11	
65B	-0.065	-0.291
Human	**0.941**	**0.827**
GPT	-0.010	-0.299
	Interval: 25	
65B	0.005	-0.077
Human	**0.657**	**0.586**
GPT	0.039	-0.186
	Interval: 35	
65B	0.017	-0.042
Human	**0.543**	**0.395**
GPT	-0.007	-0.128
	Interval: 51	
65B	0.058	0.019
Human	**0.464**	**0.334**
GPT	-0.053	-0.171

Table 2: Effect Sizes by Source and Condition. Note: r denotes the rank biserial correlation used as the effect size measure. Greater deviation from zero equates to larger disparity between registers for a particular source. Negative values indicate an opposite direction in polarity between registers for a source compared to human VMPs.

As Table 2 indicates, a discernible relationship exists with the window size employed for calculating text VMPs. It is important to note that larger window sizes correlate with identifying broader patterns within a text, whereas smaller windows are sensitive to finer-grained distinctions. A consistently higher effect size is attributed to the human source throughout the range of window sizes tested, indicating a more pronounced differentiation capability. Notably, after an initial increase from a window size of 9 to 11, the effect size for the human source gradually declines towards a window size of 51. In contrast, the 65B and GPT sources demonstrate comparatively weaker effect size strengths and fluctuate in directional tendency across varying window intervals. Comparing results with and without common words removed suggests that the more apparent register differentiation in human writing is consistent when considering both lexical and grammatical words.

4.3 Word Keyness

To answer RQ3, we extracted the top 100 *key* keywords by applying Equation (2) to distinguish between our corpora demonstrated in Table 3 for creative and Table 4 for news. Upon first inspection, there appears to be a notable propensity to use colourful language in the form of profanity, which is evident in human creative writing but absent in creative output from GPT. However, this becomes less apparent when comparing humans to 65B. Comparing the creative register, we notice a pronounced affinity towards darker thematic language expressed in human writing. Words such as: *bloody, die, torture, cry,* and *hate* are clear exemplars of this notion represented in human samples to GPT vocabulary usage. Conversely, GPT utilizes what can be described as more optimistic language, examples including: *succeeded, grateful, determined.* Some of these variations resonate between humans and 65B, but to a lesser extent. Words of interest would be aggressive or action words, such as: *threat, slammed, battle* indicating themes of conflict, whereas 65B demonstrates a polarity with humans through positive words of emotional tone, as in: *team, community, friendship*. Pulling back, we also see contractions, through the letter *d*, for human writing and when coupled with the presence of pragmatic markers *oh, uh, ah*, we can speculate on stylistic cues used by humans to signal variation in character speech, an aspect less prominent in our AI samples. Diverting our attention towards the news register, the LLMs tend to have more abstract and longer words, whereas humans tend to use more concrete and shorter words. A caveat to note here is that many of the human keywords relate to reports of events (sports results, financial results). We speculate that LLMs do not generate these (or not as much) because to do so they must start inventing specific facts. So, the keywords might reflect how LLMs are tuned to avoid levels of detail about the world that they cannot accurately emulate.

5 Conclusion

This study combined more seasoned and newer approaches for evaluating vocabulary usage between human and AI-generated texts. We noted structural differences in text sources, particularly in how VMPs respond to our research queries about discernible vocabulary patterns. Using distributional moment features like mean and standard deviation; we pinpointed statistical disparities between groups under various conditions, such as window size and vocabulary inclusion. By converting DTW distances into self-similarity measures, we observed marked differences in distributions by register for specific sources. These measurable variations underscore the distinct structural patterns of VMPs generated from different sources. Further investigation, particularly in response to RQ3, uncovered specific vocabulary that served as key indicators of thematic variations related to emotional tone. Understanding these variations can help educators and language learners select materials that best align with their learning objectives. We envision a scenario where aspects of LLM-produced text with lower mean VMPs could be combined with derived word keyness features to seek out text samples that incorporate desired vocabulary and appropriate repetition, an advantage for learning new vocabulary.

Limitations

This study takes a nuanced view of using Large Language Models (LLMs) in language learning settings. We do not oppose their use, as we recognize that there is support for such applications, and their use should align with educators' and learners' educational goals and objectives. However, we also note limitations in text selection. We acknowledge that register can be a fluid quality, and variations within a register may not be fully captured by the data used in our analysis. Moreover, although our dataset is balanced in terms of sample count, achieving a perfect balance in token length poses challenges. While truncating texts is a feasible approach, it's crucial to consider that details at the end of passages may reveal unique attributes of the sources.

AI-generated text was derived from default configurations. While adjusting parameters such as temperature or top-p could influence outcomes, we opted to examine versions which users will most likely encounter in educational settings. Our goal was to establish a baseline understanding of unaltered text production by LLMs, with plans to investigate the impact of varying parameters in future research. Gaining a deeper understanding of the production limitations of both sources can guide future research towards making LLMs more representative of human language. This insight can also effectively leverage LLMs' potential advantages.

References

Mohammad Awad AlAfnan and Siti Fatimah MohdZuki. 2023. Do Artificial Intelligence Chatbots Have a Writing Style? An Investigation into the Stylistic Features of ChatGPT-4. *Journal of Artificial Intelligence and Technology*.

Risang Baskara and Mukarto. 2023. Exploring the Implications of Chatgpt for Language Learning in Higher Education. *Indonesian Journal of English Language Teaching and Applied Linguistics*, 7(2):343–358. ERIC Number: EJ1391490.

Douglas Biber. 1987. A Textual Comparison of British and American Writing. *American Speech*, 62(2):99–119. Publisher: [Duke University Press, American Dialect Society].

Douglas Biber. 2014. Using multi-dimensional analysis to explore cross-linguistic universals of register variation. *Languages in Contrast*, 14(1):7–34.

Tanya Clement. 2013. Text Analysis, Data Mining, and Visualizations in Literary Scholarship.

Xuan-Quy Dao. 2023. Performance Comparison of Large Language Models on VNHSGE English Dataset: OpenAI ChatGPT, Microsoft Bing Chat, and Google Bard. ArXiv:2307.02288 [cs].

Jesse Egbert and Douglas Biber. 2019. Incorporating text dispersion into keyword analyses. *Corpora*, 14(1):77–104.

D. Reid Evans. 2020. On the fractal nature of complex syntax and the timescale problem. *Studies in Second Language Learning and Teaching*, 10(4):697–721.

Alex Gilmore. 2007. Authentic materials and authenticity in foreign language learning. *Language Teaching*, 40(2):97–118. Publisher: Cambridge University Press.

Stefan Th Gries. 2021. A new approach to (key) keywords analysis: Using frequency, and now also dispersion. *Research in Corpus Linguistics*, 9(2):1–33. Number: 2.

Yanzhu Guo, Guokan Shang, Michalis Vazirgiannis, and Chloé Clavel. 2023. The curious decline of linguistic diversity: Training language models on synthetic text.

Carlos Gómez-Rodríguez and Paul Williams. 2023. A Confederacy of Models: a Comprehensive Evaluation of LLMs on Creative Writing. ArXiv:2310.08433 [cs].

Mohammad Hosseini, Catherine A. Gao, David Liebovitz, Alexandre Carvalho, Faraz S. Ahmad, Yuan Luo, Ngan MacDonald, Kristi Holmes, and Abel Kho. 2023. An exploratory survey about using ChatGPT in education, healthcare, and research. Pages: 2023.03.31.23287979.

Scott Jarvis. 2013. Capturing the Diversity in Lexical Diversity. *Language Learning*, 63(s1):87–106.

Stephen Jeaco. 2023. How can we communicate (visually) what we (usually) mean by collocation and keyness?: A visual response to Gries (2022a). *Journal of Second Language Studies*, 6(1):29–60.

Ilka Kostka and Rachel Toncelli. 2023. Exploring Applications of ChatGPT to English Language Teaching: Opportunities, Challenges, and Recommendations. *Teaching English as a Second or Foreign Language–TESL-EJ*, 27(3).

Haidee Kruger and Bertus Van Rooy. 2018. Register variation in written contact varieties of English: A multidimensional analysis. *English World-Wide. A Journal of Varieties of English*, 39(2):214–242.

Bruce W. Lee and Jason Lee. 2023. LFTK: Handcrafted Features in Computational Linguistics. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 1–19, Toronto, Canada. Association for Computational Linguistics.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2023. Deepfake text detection in the wild.

Carl H. Lubba, Sarab S. Sethi, Philip Knaute, Simon R. Schultz, Ben D. Fulcher, and Nick S. Jones. 2019. catch22: CAnonical Time-series CHaracteristics. *Data Mining and Knowledge Discovery*, 33(6):1821–1852.

John McKenny. 2003. Seeing the wood and the trees: Reconciling findings from discourse and lexical analysis. In *Paper at Corpus Linguistics 2003 Conference, University of Lancaster. University Centre for Computer Corpus Research on Language (UCREL). Technical Papers*, volume 16.

Jim Meyer and Brendan Cooney. 1994. The paragraph: Towards a richer understanding. *Work Papers of the Summer Institute of Linguistics, University of North Dakota Session*, 38(1).

David Mitchell. 2015. Type-token models: a comparative study. *Journal of Quantitative Linguistics*, 22(1):1–21. Publisher: Routledge _eprint: https://doi.org/10.1080/09296174.2014.974456.

Keith Morrow. 1977. Authentic texts and esp. *English for specific purposes*, 13:17.

I. Nation. 2006. How Large a Vocabulary is Needed For Reading and Listening? *The Canadian Modern Language Review*, 63(1):59–82. Publisher: University of Toronto Press.

I. S. P. Nation. 2011. Research into practice: Vocabulary. *Language Teaching*, 44(4):529–539.

Partha Pratim Ray. 2023. ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems*, 3:121–154.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2023. The curse of recursion: Training on generated data makes models forget.

Michael Stubbs. 2006. READING D: Exploring 'Eveline' with computational methods. In Sharon Goodman and Kieran O'Halloran, editors, *The art of English: literary creativity*, pages 138–144. Palgrave Macmillan ; In Association with the Open University, Basingstoke [England] ; New York : Milton Keynes, UK. OCLC: 63705884.

Silvia Vaccino-Salvadore. 2023. Exploring the Ethical Dimensions of Using ChatGPT in Language Learning and Beyond. *Languages*, 8(3):191. Publisher: MDPI AG.

Gilbert Youmans. 1990. Measuring Lexical Style and Competence: The Type-Token Vocabulary Curve. Accepted: 2009-02-05T21:48:19Z Publisher: Northern Illinois University.

Gilbert Youmans. 1991. A New Tool for Discourse Analysis: The Vocabulary-Management Profile. *Language*, 67(4):763–789. Publisher: Linguistic Society of America.

Gilbert Youmans. 1994. The Vocabulary-Management Profile: Two Stories by William Faulkner. *Empirical Studies of the Arts*, 12(2):113–130. Publisher: SAGE Publications Inc.

Zorica Đurović. 2023. Frequency or Keyness? *Lexikos*, 33.

A Appendix

Algorithm 1 Calculate Dispersion of a *word* in a Corpus

```
Require: A corpus divided into $N$ parts, the word in question, $N \geq 1$
Ensure: Dispersion value of the word in range [0,1]
Let $N$ be the number of parts, $N = 10$
Initialize array $F$ to store the frequency $f_i$ of the word in each part $i$
Initialize array $S$ to store the size $s_i$ of each part $i$
Initialize $D_{KL}$ to 0
for $i = 1$ to $N$ do

    $p_i \leftarrow \frac{f_i}{\sum_{j=1}^{N} f_j}$

    $q_i \leftarrow \frac{s_i}{\sum_{j=1}^{N} s_j}$

    if $p_i > 0$ then
        $D_{KL} \leftarrow D_{KL} + p_i \times \log_2\left(\frac{p_i}{q_i}\right)$
    end if
end for
Dispersion $\leftarrow 1 - e^{-D_{KL}}$
return Dispersion =0
```

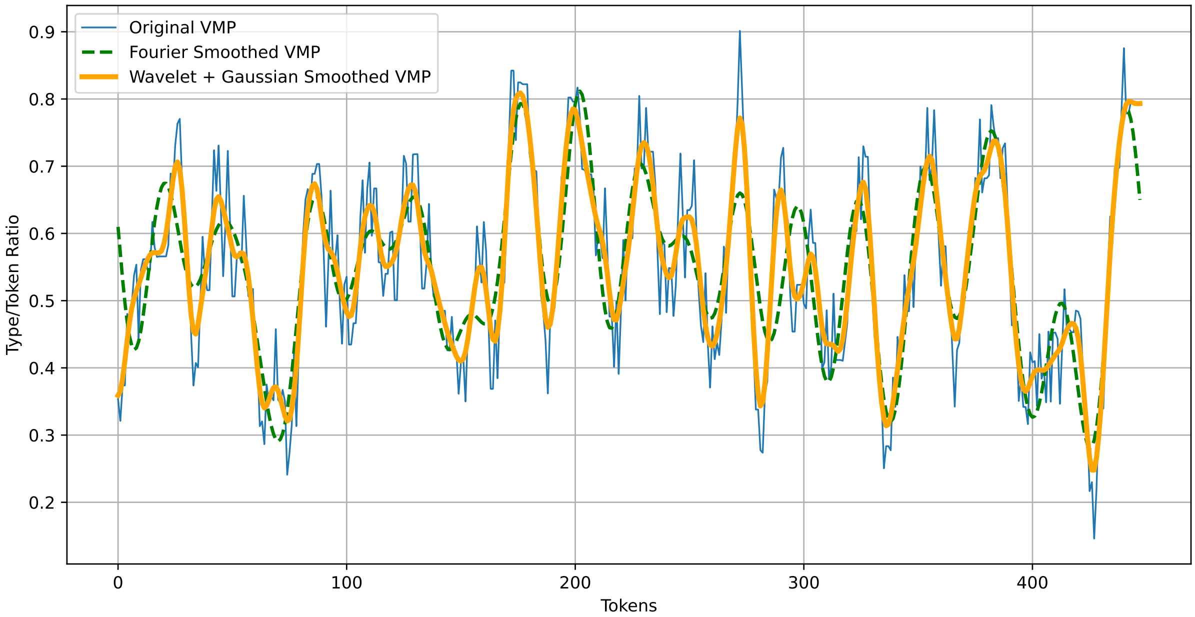


Figure 5: Smoothing transformation of news text sample from a human source for commonNo vocabulary condition with a window delta value 9. The blue solid line represents the original VMP data exhibiting natural variability and noise. The green dashed line shows the VMP data after Fourier transform-based smoothing, which reduces high-frequency fluctuations while preserving the main signal trend. The orange solid line, bolder for emphasis, displays the VMP data subjected to a two-stage smoothing process involving wavelet denoising followed by Gaussian smoothing, offering a balance between noise reduction and signal integrity preservation.

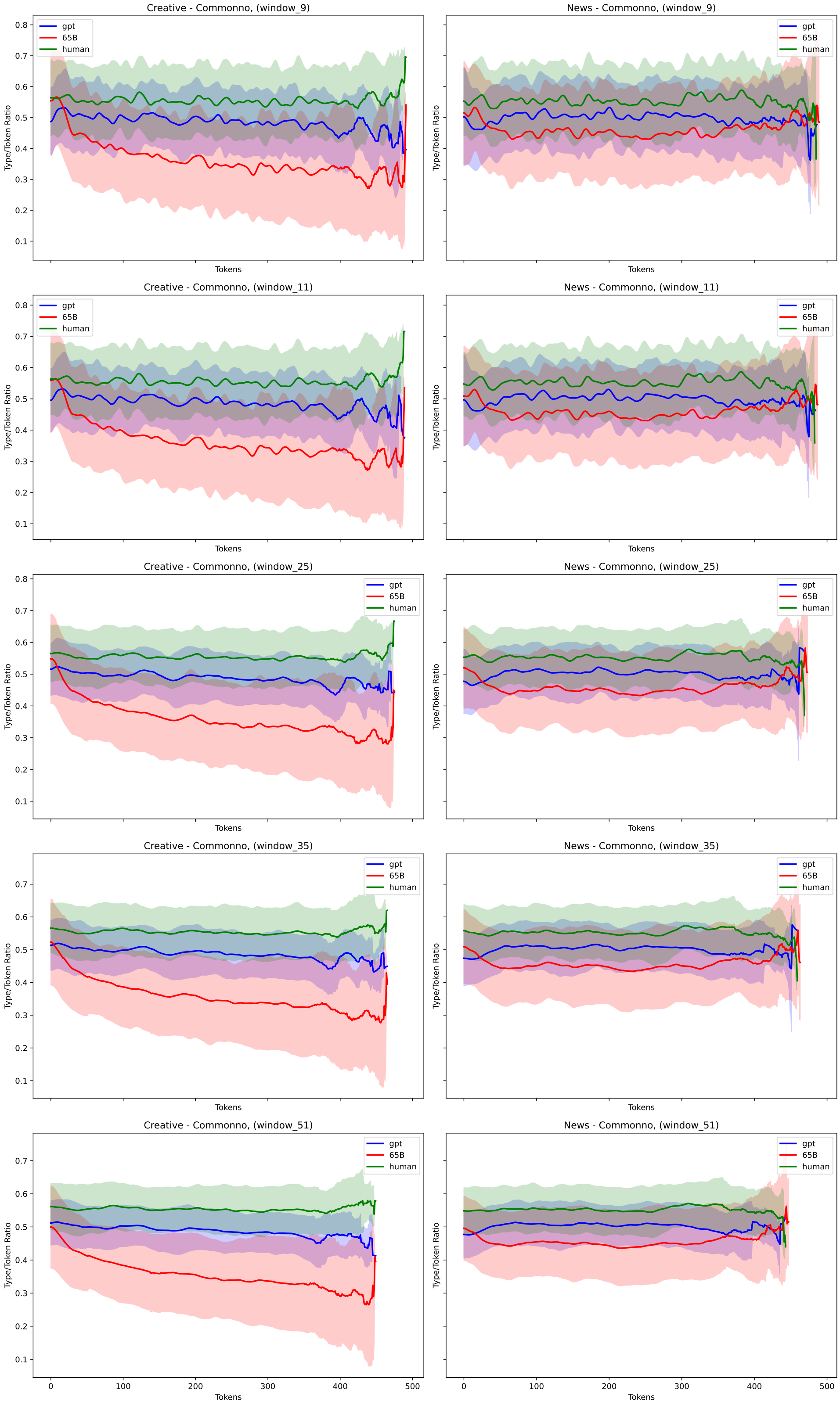


Figure 6: VMP commonNo

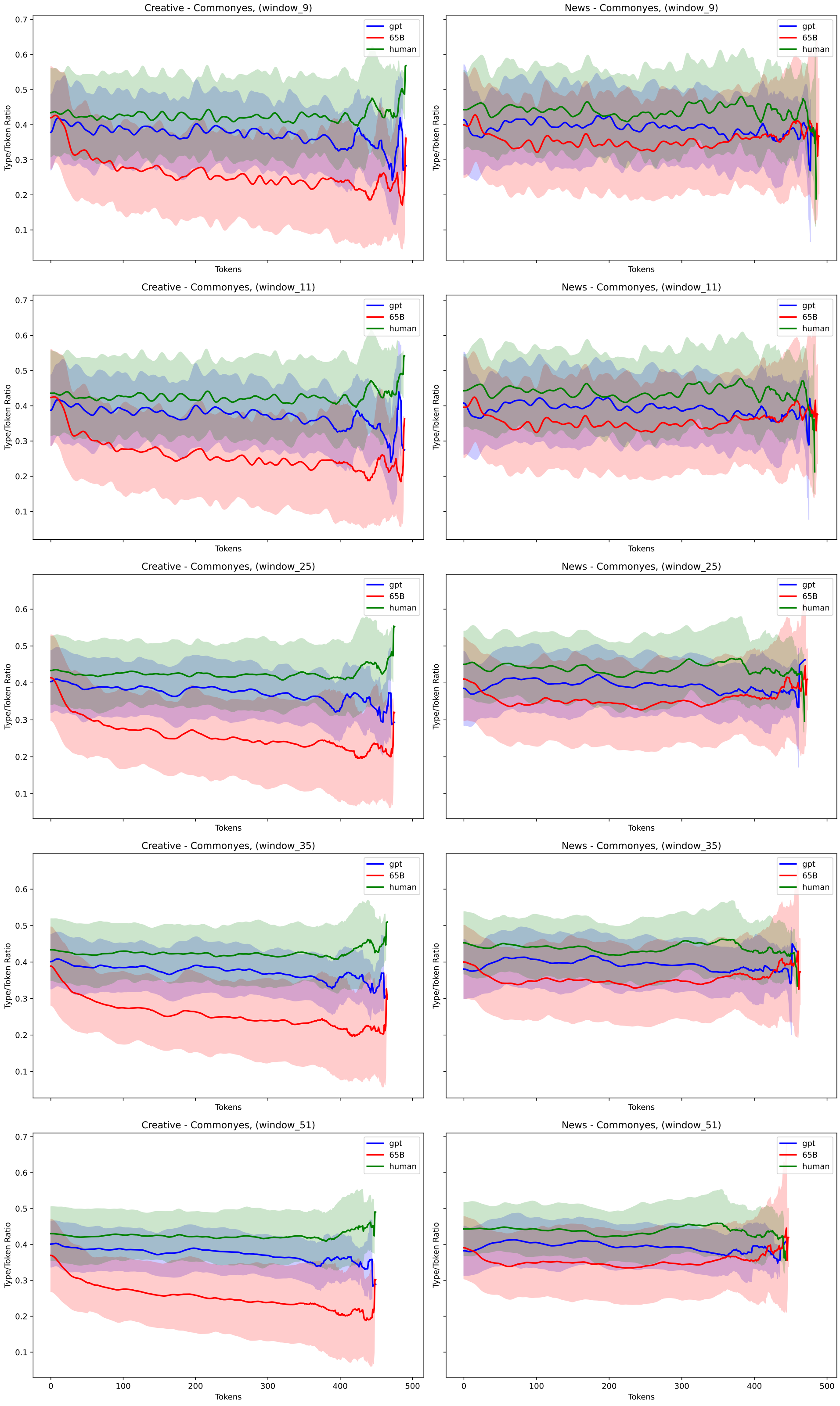


Figure 7: VMP commonYes

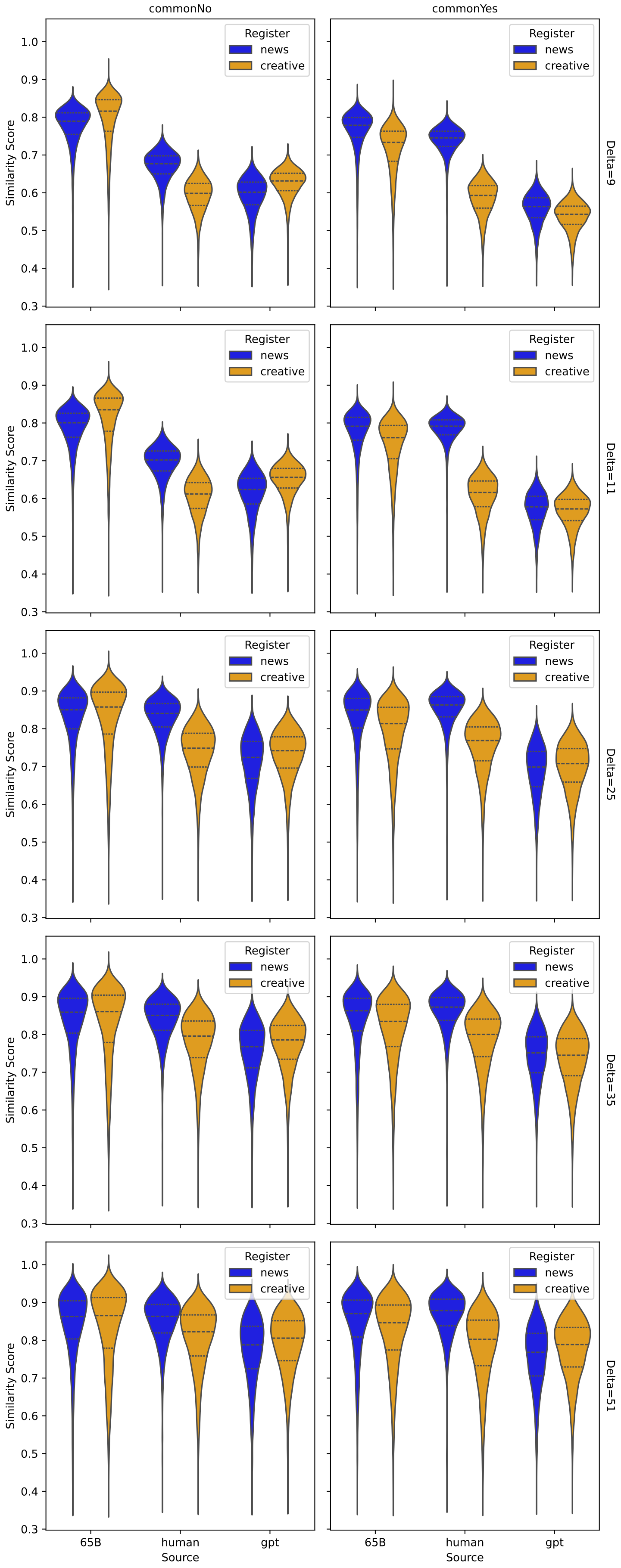


Figure 8: Self-similarity distribution plots of different sources for varying window sizes

Top 100 *Key* Keywords for Creative Register
Creative Target-human
ll, looks, mouth, oh, wo, says, pretty, couple, seems, anyway, fuck, hell, stuff, damn, office, ca, shit, paper, women, bastard, gets, picture, shouted, fingers, shoulder, definitely, ate, ah, guess, please, d, hate, knows, orange, sit, god, cry, torture, fault, die, na, edit, click, direction, deserve, relationship, breathing, normally, honest, flesh, son, jacket, eight, suppose, send, chair, cabinet, y, till, smoke, pile, armor, reaching, inevitable, starving, kicked, fed, hated, realise, shouting, chin, somewhere, hanging, kinda, scratch, gon, muster, nope, alcohol, bloody, blood, roll, slammed, dollars, yearold, decent, lights, accent, cheek, sits, bathroom, gotten, deserved, asleep, tear, writing, uh, literally, hall, obviously
Creative Reference-gpt
named, fascinated, grateful, tirelessly, determined, shared, excitement, significant, accepted, completed, relieved, consequences, lily, overjoyed, hesitant, sophie, practicing, including, respect, protect, differences, overwhelmed, intricate, eagerly, welcomed, skeptical, traveled, thrilled, unique, hugged, detail, opportunity, villagers, chatted, achieved, gathering, longed, mattered, approach, spreading, genuinely, expert, deserted, focused, catching, colony, choosing, importance, promising, mesmerized, frustrated, defend, insects, grew, noticed, rush, impressed, series, challenging, thrill, rebuild, value, succeeded, dense, behavior, lush, warriors, puzzle, intrigued, became, lilys, alex, determination, jacks, granted, technology, weapons, crops, team, gaining, decision, insight, peculiar, crucial, particularly, tool, dire, mortal, practiced, equally, routines, facility, frustration, mustered, grueling, industry, forests, judged, impending, sunny
Creative Target-human
deep, soul, seemed, slightly, perhaps, shit, powers, bastard, rise, ago, warm, address, count, swear, absolutely, further, thousand, though, impact, torture, odd, discovered, whenever, frozen, million, heading, normally, existence, sea, carry, appeared, necessary, battle, reality, flesh, definitely, century, similar, entered, jacket, eight, seven, data, cabinet, y, rushing, till, armor, dull, reaching, relief, inevitable, starving, clear, kicked, actual, brings, realise, space, souls, instant, blanket, kinda, smart, slow, muster, tightly, placed, causing, hands, somehow, threat, slammed, progress, landed, pressed, surely, stars, gold, silly, wet, bodies, gun, seeking, uh, advanced, literally, humanity, hundred, faster, advance, officers, pure, masters, leader, disgusting, intelligence, breath, particular, master
Creative Reference-65B
ruin, example, protect, couch, posted, pick, neighbors, labels, blow, upset, woods, scary, particularly, oven, writer, treat, ridiculous, suggested, jealousy, department, services, talks, relieved, pray, cleaned, react, financial, candy, persons, october, horny, depressed, glad, policy, music, thankful, hmm, levels, recover, ages, accomplish, cream, creepy, dads, feeding, filed, necklace, repairing, hugs, easter, nerve, ideas, liable, operating, nobodys, including, areas, sentences, hugged, blowing, kissing, cases, acting, concentrate, shadowy, rules, teaches, cooking, player, fund, jump, students, widened, filing, respect, christian, mix, investigate, explaining, curb, tubes, rural, recipe, airport, costs, fishing, backyard, lakes, tragic, statements, stabbing, expressing, crook, rode, sisters, borrowed, sobs, todays, amazon, dance

Table 3: Keyword Summary for Creative Register

Top 100 *Key* Keywords for News Register
News Target-human
psm, died, mr, main, ps, wednesday, parents, probably, d, told, added, near, spokesman, travelling, happened, eight, deputy, monday, thursday, mrs, talk, playback, radio, ms, morning, huge, apparently, march, records, single, either, chief, county, editor, weather, professor, captain, consumer, psbn, appeared, going, boss, refused, go, me, april, rangers, labours, accepted, twitter, crown, strongly, det, backing, possibly, internationally, brother, linked, partner, insurance, mps, achieved, communications, pictures, advised, loan, might, tv, recognised, flat, insisted, brilliant, absolutely, evening, nice, afford, strikes, afternoon, voted, sat, door, targets, staged, chris, obviously, innings, broke, estimates, bst, troops, injury, stephen, christmas, jail, four, pretty, pupils, stopped, scottish, ibrox
News Reference-gpt
conclusion, importance, culture, ultimately, practices, shape, behavior, risks, attention, argue, criticized, navigate, experiences, essential, efforts, deeply, traditional, stranger, dynamic, impossible, consumption, highlighting, thrilled, inspire, accountability, tech, ceo, arguing, unique, individuals, growing, ability, promising, towards, noted, alike, remains, organization, volatile, diagnosed, defense, resilient, proven, likes, uncertain, inspiration, unexpected, combat, effects, tasked, observers, examples, dedicated, opponents, ensuring, organized, guidance, topic, transition, responsibility, stable, handling, dedication, tirelessly, investigations, discrimination, muchneeded, implement, accessible, gender, controversy, significant, highprofile, emissions, takes, engaging, collaboration, transparent, remarks, uncertainty, recognized, laws, disputes, scandal, wellknown, ethical, achieving, cultural, create, spread, pandemic, equalizer, caution, ramp, cautious, component, effectively, scandals, strain, disrupt
News Target-human
speaking, revealed, troops, fans, warned, psm, regular, followed, pitch, deputy, september, ps, powers, radio, might, adding, losing, deals, prove, parent, eventually, independently, suggested, average, quarter, premiership, aged, rising, rugby, wanted, marks, african, bbc, labours, chose, praised, latter, backing, armed, internationally, monthly, eyes, sheffield, historic, loan, disruption, cold, unbeaten, recognised, flat, insisted, crowd, outcome, mistake, evening, strikes, proper, staged, obviously, operation, retain, complex, standing, celtic, lose, ownership, employers, games, favour, nottingham, sundays, euro, sparked, ali, stake, commit, mile, mutual, responding, dealt, length, appalling, militants, sit, defended, institution, indicated, contributed, automatically, quoted, rebuild, clearly, southeast, broken, subsequently, scores, ira, formal, cancelled, sofa
News Reference-65B
researchers, applications, center, delivers, ceo, base, method, mexico, photos, billion, developers, promising, organized, manifesto, awarded, apples, amazon, operated, organization, developer, deeply, development, effects, displayed, capabilities, export, author, episode, distributed, browser, stimulate, chapter, determine, forget, netherlands, flag, detective, object, restrictions, ties, caring, surveillance, spread, organizations, manipulate, dedicated, residential, factories, integrated, cruz, perfect, entry, tagged, takes, earth, gentleman, guy, cultural, approved, library, stable, apple, ability, trained, illinois, patterns, tags, updated, federation, consulting, vulnerability, sensitive, acquisition, useful, crew, airports, implemented, physics, tool, humans, interact, algorithms, valley, please, virtual, algorithm, materials, located, threats, historical, tools, fuel, australias, experiences, movies, manage, afraid, experiment, string, asks.

Table 4: Keyword Summary for News Register

Exploring LLM Prompting Strategies for Joint Essay Scoring and Feedback Generation

Maja Stahl
Leibniz University Hannover
m.stahl@ai.uni-hannover.de

Leon Biermann
Leibniz University Hannover
leon.biermann@stud.uni-hannover.de

Andreas Nehring
Leibniz University Hannover
nehring@idn.uni-hannover.de

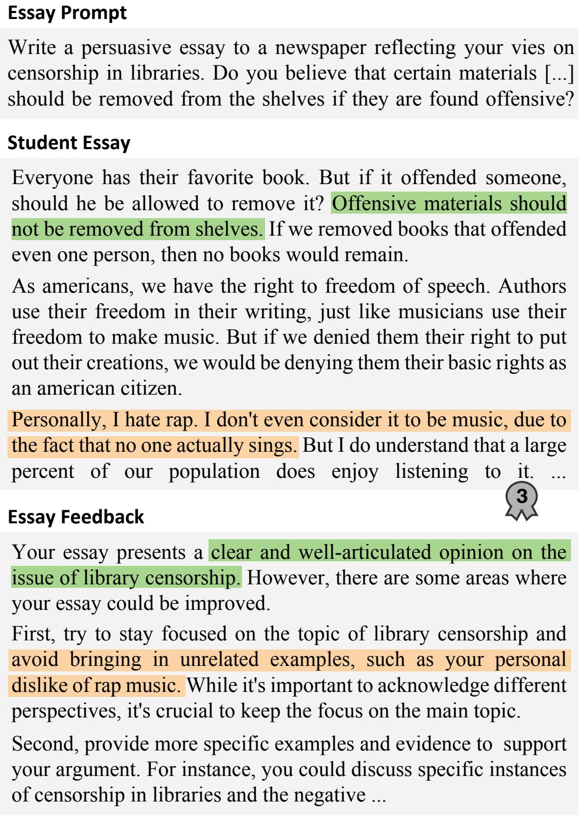
Henning Wachsmuth
Leibniz University Hannover
h.wachsmuth@ai.uni-hannover.de

Abstract

Individual feedback can help students improve their essay writing skills. However, the manual effort required to provide such feedback limits individualization in practice. Automatically-generated essay feedback may serve as an alternative to guide students at their own pace, convenience, and desired frequency. Large language models (LLMs) have demonstrated strong performance in generating coherent and contextually relevant text. Yet, their ability to provide helpful essay feedback is unclear. This work explores several prompting strategies for LLM-based zero-shot and few-shot generation of essay feedback. Inspired by Chain-of-Thought prompting, we study how and to what extent automated essay scoring (AES) can benefit the quality of generated feedback. We evaluate both the AES performance that LLMs can achieve with prompting only and the helpfulness of the generated essay feedback. Our results suggest that tackling AES and feedback generation jointly improves AES performance. However, while our manual evaluation emphasizes the quality of the generated essay feedback, the impact of essay scoring on the generated feedback remains low ultimately.

1 Introduction

Essay writing is a valuable skill that empowers students to communicate effectively, to think critically, and to express their opinions clearly (Bangert-Drowns et al., 2004; Liu et al., 2014; Schendel and Tolmie, 2016). However, learning to write essays of high quality can be challenging (Peloghitis, 2017; Febriani, 2022). Individual feedback to essays is thus critical, so students may recognize and overcome their weaknesses (Riddell, 2015). However, manually writing essay feedback is time- and labor-intensive. Given educators' limited time, this often renders real individual feedback infeasible in practice. As an alternative, automated essay writing support could benefit students by offering guidance at their own pace and convenience.



Essay Prompt

Write a persuasive essay to a newspaper reflecting your vies on censorship in libraries. Do you believe that certain materials [...] should be removed from the shelves if they are found offensive?

Student Essay

Everyone has their favorite book. But if it offended someone, should he be allowed to remove it? Offensive materials should not be removed from shelves. If we removed books that offended even one person, then no books would remain.

As americans, we have the right to freedom of speech. Authors use their freedom in their writing, just like musicians use their freedom to make music. But if we denied them their right to put out their creations, we would be denying them their basic rights as an american citizen.

Personally, I hate rap. I don't even consider it to be music, due to the fact that no one actually sings. But I do understand that a large percent of our population does enjoy listening to it. ...

3

Essay Feedback

Your essay presents a clear and well-articulated opinion on the issue of library censorship. However, there are some areas where your essay could be improved.

First, try to stay focused on the topic of library censorship and avoid bringing in unrelated examples, such as your personal dislike of rap music. While it's important to acknowledge different perspectives, it's crucial to keep the focus on the main topic.

Second, provide more specific examples and evidence to support your argument. For instance, you could discuss specific instances of censorship in libraries and the negative ...

Figure 1: Exemplary student essay on library censorship from the ASAP dataset (Hamner et al., 2012) along with feedback and essay score generated by one of the methods evaluated in this paper. Explicit connections of the feedback to essay parts are color-coded.

Towards supporting essay writing automatically, much research has focused on quality assessment, mostly in the form of *automated essay scoring* (AES; Ke and Ng, 2019; Ramesh and Sanampudi, 2022). Most existing AES approaches perform a holistic scoring, summarizing the quality of an essay with a single score (Alikaniotis et al., 2016; Vajjala, 2018; Uto et al., 2020; Wang et al., 2023). Alternatively, specific essay quality aspects may be assessed, such as coherence (Li et al., 2018; Farag et al., 2018), grammar (Ajit Tambe and Kulkarni, 2022), and organization (Persing et al., 2010;

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 283–298
June 20, 2024 ©2024 Association for Computational Linguistics

Rahimi et al., 2015). While AES helps assess essay quality and monitor writing skill progress, most approaches cannot explain why a score was predicted, nor guide the student in improving the essay.

Another prominent line of research towards writing support focuses on identifying and correcting grammatical errors (Imamura et al., 2012; Bryant et al., 2017; Rozovskaya and Roth, 2019; Grundkiewicz et al., 2019). However, studies in educational research show that computer-based learning systems lead to higher learning outcomes if elaborated feedback is provided that provides explanations instead of only pointing to errors or providing the solution (Van der Kleij et al., 2015). Therefore, Nagata (2019) introduced the task of *feedback comment generation* in NLP: Given a learner text with a grammatical error, automatically generate a comment with hints and explanations to guide their correction process. Song et al. (2023) extended this task by generating explanations for a broader range of grammatical error types using large language models (LLMs). However, these tasks operate only on the sentence level and are limited to grammatical errors. Generating feedback on the essay level by addressing not only grammatical errors but the essay as a whole remains relatively unexplored.

To foster research in this direction, we tackle the task of *essay feedback generation*: Given a student essay, automatically generate textual feedback that helps students improve their essays. An example is shown in Figure 1. Building on the strong abilities of LLMs in many text-generation tasks, this work examines how well LLMs can generate essay feedback by exploring various prompting strategies in zero- and few-shot settings. Inspired by Chain-of-Thought prompting (Wei et al., 2022), we study whether AES can benefit the performance of essay feedback generation and vice versa.

Our experiments suggest that generating essay feedback by explaining the predicted essay score improves the scoring performance on the widely-used ASAP dataset (Hamner et al., 2012). For essay feedback generation, we deem helpfulness to be the most important quality criterion. Helpful essay feedback should point out and explain mistakes made in an essay in a precise and easy way for students to understand (Shute, 2008; Hattie and Timperley, 2007). We evaluate the helpfulness automatically and manually. Due to the lack of ground-truth essay feedback, we propose using LLMs to automatically judge the essay feedback's helpfulness, which turns out to correlate well with human helpfulness judgments. Our manual evaluation also reveals that the generated essay feedback is deemed helpful for students to improve their essay writing skills. However, the impact of scoring the essay remains low ultimately. Altogether, this paper's main contributions are:

- A comparison of several LLM prompting strategies for automated essay scoring
- An approach and task-specific automatic evaluation strategy for essay feedback generation using LLM prompting
- Empirical insights into the influence of automated essay scoring on generating essay feedback and vice versa[1]

2 Related Work

Essay writing is a central task in education to evaluate various skills of students, including logical thinking, critical reasoning, and creativity (Liu et al., 2014; Schendel and Tolmie, 2016). However, manual essay grading is time-consuming and not always consistent within and across raters (Kassim, 2011; Eckes, 2015). Automated essay scoring (AES) aims to alleviate these issues, reducing the effort of graders and, ideally, making grading more consistent and reliable (Ke and Ng, 2019; Uto, 2021; Ramesh and Sanampudi, 2022).

While extensive research exists on AES (Ke and Ng, 2019; Ramesh and Sanampudi, 2022), assessing all important quality aspects (known as *traits*), including the relevance of an essay's content to the prompt, the development of ideas, cohesion, coherence, and more remains challenging (Ramesh and Sanampudi, 2022). Only few works focus on scoring multiple traits at once (Mathias and Bhattacharyya, 2020; Hussein et al., 2020). Instead, the majority of AES research targets holistic essay scoring, that is, summarizing the essay quality in a single score (Alikaniotis et al., 2016; Cozma et al., 2018; Vajjala, 2018; Wang et al., 2023).

State-of-the-art approaches to AES can be divided by their use of the available data into full-data and few- or zero-shot settings (Tao et al., 2022). In the full-data setting, where all labeled data is used for training, most approaches fine-tune pretrained language models, such as BERT (Devlin et al., 2019). Yang et al. (2020) proposed solving the task

[1] The code used for our experiments can be found under https://github.com/webis-de/BEA-24.

by combining essay scoring and essay ranking, fine-tuning BERT using multiple losses simultaneously. Extending this idea, Xie et al. (2022) combined regression and ranking into a single loss. Rather than fine-tuning a language model, Tao et al. (2022) designed two self-supervised constraints for learning a multi-layer embedding, which prepends the input to a frozen BERT model. They evaluate their approach in the full-data and one-shot setting, outperforming a fine-tuned BERT in the latter. To explore the potential of large language models (LLMs), Mizumoto and Eguchi (2023) prompted GPT-3.5 to score the student essays from the TOEFL11 dataset (Blanchard et al., 2013) in a zero-shot setting, indicating promising scoring performance.

The most straightforward way to provide more detailed feedback for an essay than a holistic score is trait scoring (Jong et al., 2023), that is, to evaluate an essay for different quality aspects. However, the reasoning behind an assigned trait score usually remains unknown to the student. Therefore, Kumar and Boulanger (2020) adopted explainability methods to explain how input features to an AES system influence the trait scores for an essay. While this provides more insights, the pedagogical quality and impact on writing performance remain questionable if no feedback is given together with the scores (Kumar and Boulanger, 2020).

Specific feedback generation tasks have been addressed in educational NLP. Nagata (2019) introduced *feedback comment generation* to explain grammatical errors to a learner on the sentence level. This task has been tackled by combining retrieval and text generation (Hanawa et al., 2021; Ihori et al., 2023), by identifying different feedback types (Stahl and Wachsmuth, 2023), by augmenting the dataset (Babakov et al., 2023; Behzad et al., 2023), and by correcting the error (Jimichi et al., 2023; Koyama et al., 2023), all using fine-tuned language models. For a wider range of grammatical error types, Song et al. (2023) used the LLM GPT-4 to first identify the necessary corrective edit before generating a grammar error explanation using one-shot prompting. In the educational domain, Meyer et al. (2024) showed that LLM-based writing feedback, generated using a single handcrafted prompt, positively impacts students' text revisions, motivation, and positive emotions.

So far, however, the generation of textual feedback on complete student essays has, to our knowledge, received very little attention. All generation approaches mentioned above operate on the sentence level and explain grammatical errors only, while our work aims to address all aspects of student essays that may need improvement. The only other work on essay feedback generation tackled the task using Chain-of-Thought prompting using zero-shot learning (Han et al., 2023). The resulting feedback was deemed to be more helpful than the feedback generated using standard prompting, as evaluated by humans.

Motivated by these promising results and the positive effects of LLM-based writing feedback on students, we go beyond previous work by comparing the effectiveness of different prompting strategies for essay feedback generation. We also study how and to what extent AES can benefit essay feedback generation (and vice versa) by addressing the tasks jointly. Following the educational literature on feedback, we aim to generate essay feedback that is specific and elaborate (Shute, 2008) while assessing the current state and instructing on how to improve to achieve the goals (Hattie and Timperley, 2007).

3 Approach

This section describes our approach to essay feedback generation. We propose to tackle essay scoring and feedback generation jointly in order to study how and to what extent AES can benefit essay feedback generation and vice versa. By comparing different prompting strategies for large language models (LLMs), we explore how well the tasks can be solved using in-context learning.

In particular, we test different prompting strategies by systematically varying three main aspects of the prompts, as visualized in Figure 2: (a) the *prompt pattern*, which defines the context and layout; (b) the *task instruction type*, which sets the ordering and phrasing of the tasks to be tackled; and (c) the *in-context learning* approach, which specifies the number of examples provided.

3.1 Prompt Patterns

We compare two different kinds of prompt patterns, which define the context and format of the prompt: (i) a *base pattern* and (ii) different *persona patterns*. All prompt patterns are displayed in Table 1.

Base Pattern The base pattern simply gives the general context and defines the layout and order in which the *essay prompt* (i.e., the task given to the learner writing the essay), the *task instruction*, as

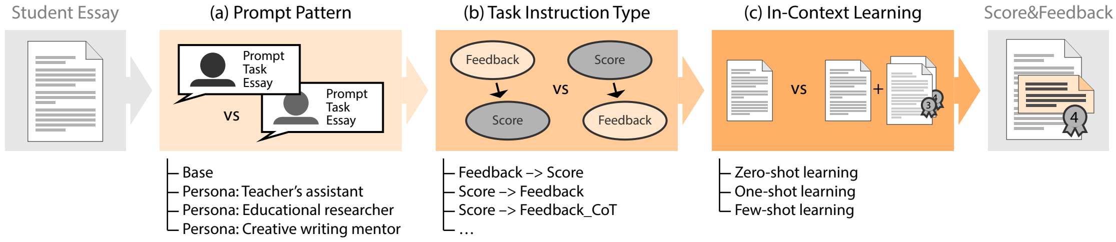


Figure 2: Overview of the main points of variation in our approach to predict a score and to generate feedback for a student essay: (a) Prompt pattern: Use of the base pattern or persona-specific pattern; (b) Task instruction type: Tasks to be tackled and their ordering; (c) In-context learning approach: Number of examples to learn from.

Base: You are given an essay written by a student and the corresponding prompt for the 7th to 10th grade student. #### Prompt: "{*essay_prompt*}" ### Task: {*task_instruction*} #### Student essay: "{*essay*}"
Teacher's Assistant: Imagine you are a teacher's assistant in a middle school tasked with reviewing a 7th to 10th grade student's essay. You have the essay and the prompt that was given to the student. #### Original Prompt Provided to Student: "{*essay_prom.*}" ### Review Task: {*task_instruction*} #### Student's Essay for Review: "{*essay*}"
Educational Researcher: You are part of an educational research team analyzing the writing skills of students in grades 7 to 10. You have been given a student's essay and the prompt they responded to. #### Essay Prompt: "{*essay_prompt*}" ### Analysis Task: {*task_instruction*} #### Analyzed Student Essay: "{*essay*}"
Creative Writing Mentor: You are a creative writing mentor evaluating a piece written by a student in grades 7 to 10. The student's work is based on a specific prompt. #### Creative Prompt Given: "{*essay_prompt*}" ### Critique Instructions: {*task_instruction*} #### Student's Creative Piece: "{*essay*}"

Table 1: Prompt patterns: Base pattern and all persona patterns. Brackets indicate placeholders that are filled respectively during the experiments. We removed model-specific pre-/suffixes and line breaks for illustration.

defined by the used task instruction type, and the current learner *essay* will be presented to the model. All inputs are indicated by markdown headings.

Persona Patterns These prompt patterns are inspired by persona prompting (White et al., 2023), giving the LLM a persona or role to play when generating output. This aims to implicitly define the expected type of output. For our task, we compare the three personas, namely, *teacher's assistant*, *educational researcher*, and *creative writing mentor*, by altering the context given in the prompt pattern.

3.2 Task Instruction Types

The task instruction type defines the tasks to be tackled along with their ordering. We differentiate between tackling (i) only essay *scoring*, (ii) essay *scoring and feedback* generation, and (iii) only essay *feedback* generation. This way, we can measure the influence that essay scoring has on feedback generation, and vice versa. We explore the following task instruction types for our tasks:

- *Scoring*. Instruct to score the student essay on a given score range. This serves as a baseline for assessing the essay scoring performance.
- *Feedback*. Instruct to generate essay feedback for the student writer. This serves as a baseline for assessing the feedback performance.
- *Scoring→Feedback*. Instruct to score the essay and then generate feedback for the student writer. This measures the influence of essay scoring on the feedback performance.
- *Feedback→Scoring*. Instructs to first generate feedback before scoring the essay. This evaluates whether feedback generation helps to predict the correct essay score.
- *Scoring→Feedback_CoT*. Instruct to score the essay and to then generate feedback using zero-shot Chain-of-Thought (CoT) prompting, that is, to add the phrase "Let's think step by step.", which has been shown to increase LLM's reasoning performance (Kojima et al., 2022). This might benefit the reasoning needed in feedback generation.
- *Feedback_dCoT→Scoring*. Instruct to first analyze the essay quality using the rubric, to then generate feedback, and to finally score the essay. This is a more detailed variation of CoT that provides task-specific steps to follow before arriving at the final essay score.

Score	Description
3	The response demonstrates an understanding of the complexities of the text. – Addresses the demands of the question – Uses expressed and implied information from the text – Clarifies and extends understanding beyond the literal
2	The response demonstrates a partial or literal understanding of the text. – Addresses the demands of the question, although may not develop all parts equally – Uses some expressed or implied information from the text to demonstrate understanding – May not fully connect the support to a conclusion or assertion made about the text(s)
1	The response shows evidence of a minimal understanding of the text. – May show evidence that some meaning has been derived from the text – May indicate a misreading of the text or the question – May lack information or explanation to support an understanding of the text in relation to the question
0	The response is completely irrelevant or incorrect, or there is no response.

Table 2: Exemplary rubric from essay set 3 of the ASAP dataset (Hamner et al., 2012). The rubrics are provided as additional information within the task instructions.

- *Scoring→Explanation*. Instruct to score the essay and to then generate an explanation for the predicted score. This explores whether score explanations as a form of feedback relate to asking for essay feedback specifically.
- *Explanation→Scoring*. Instruct to analyze the essay, to then first generate an explanation for an essay score that, in turn, should be generated at the end. This avoids that the LLM predicts an incorrect score and then generates an explanation justifying the incorrect score, as observed by Ye and Durrett (2022).

Task instructions for essay scoring provide the *scoring range* that should be used, while those for feedback generation provide the *rubric*, that is, guidelines including a short description for essays of each quality level and typical elements of such. An exemplary *rubric* can be seen in Table 2.

Since the performance of LLMs is sensitive to the exact wording of a prompt (Leidinger et al., 2023), we create a total of four *task instructions* for each task instruction type by instructing ChatGPT (OpenAI, 2023) to generate three paraphrases of each initial, manually written task instruction. Examples of the latter can be seen in Table 3. We provide all task instructions in Appendix A.

Scoring: Given this essay that was written for the given prompt, grade the essay using those ranges: {*scoring_range*}.

Feedback: Analyze the given essay using the following rubric: {*rubric*}. Provide comprehensive feedback for the student that helps them to achieve better grades in the future.

Scoring→Feedback: Grade the given essay using the following rubric: {*rubric*}. Use those score ranges: {*scoring_range*}. Provide comprehensive feedback for the student that helps them to achieve better grades in the future.

Feedback_dCoT→Scoring: Analyze the given essay using the following rubric and give helpful feedback to the student: {*rubric*}. Use those score ranges: {*scoring_range*}. Let's think step by step. First, analyze the quality of the essay in terms of the given rubric. Then, give feedback to the student that explains their mistakes and errors and additionally gives them tips to avoid them in the future. As a final step, output the score at the end.

Scoring→Explanation: Grade the given essay using the following rubric: {*rubric*}. Use those score ranges: {*scoring_range*}. Provide an explanation for your score as well.

Table 3: Task instruction types: Examples of the initial, manually written task instructions for five types. Brackets indicate placeholders that are filled with the respective information during the experiments.

One-shot Example: Essay: "{*essay*}"

Reasoning: This is a minimally-developed response with inadequate support and detail. The writer takes the position that computers can be harmful to the eyes and then addresses eye damage to three groups of people (kids, teens, adults). A few specific details are included (sensitive eyes, MySpace), but elaboration is minimal. Some organization is demonstrated but few transitions are used. Overall, the response is sufficiently developed to move into the score point '3' range.

Scores: {Overall: 3}

Table 4: One-shot example consisting of a student essay, a manually written score justification, and the assigned score. The data is taken from the scoring guidelines for essay set 1 of the ASAP dataset (Hamner et al., 2012).

3.3 In-Context Learning

As final point of variation of our approach, we explore how providing one or multiple exemplary essays, together with their score and a reasoning for the score, helps with essay scoring and feedback generation. The data comes from additional material given to human raters. We argue that the reasoning of the score may help with essay scoring, but could also be seen as a form of feedback and may benefit that task as well. We compare (i) *zero-shot*, (ii) *one-shot*, and (iii) *few-shot* learning.

For one-shot, we randomly select an essay with a medium score, as the one in Table 4. For few-shot, we first randomly select examples among the essays with the best and worst scores before covering the

Pattern	Essay Set								
	1	2	3	4	5	6	7	8	Mean
Base	.495	.532	.405	.495	.497	.601	.436	.377	.480
TA	**.536**	**.603**	.408	.499	.512	**.625**	.443	.439	.508
ER	.436	.554	**.460**	**.560**	**.553**	.620	.418	**.467**	**.509**
CWM	.484	.588	.382	.434	.507	.596	**.471**	.352	.477

Table 5: Essay scoring results: Average QWK over all task instructions using zero-shot learning for each prompt pattern: base, teacher's assistant (TA), educational researcher (ER), and creative writing mentor (CWM). We report the performance for each of the eight essay sets as well as the mean QWK over all sets.

other scores. Due to the limited context length, we restrict the prompt to 5,120 characters and select as many examples that fit this limitation as possible.[2]

4 Data

Multiple AES datasets are available, with the Automated Student Assessment Prize's (ASAP) dataset (Hamner et al., 2012) being the most widely used. It comprises 12,980 essays written by school students in grades 7 to 10. All essays were scored manually by two raters. The essays are divided into eight essay sets. The essay sets differ by the essay prompt, i.e., the task description they were written for, the scoring range, and the rubric used by the raters as annotation guidelines. The rubrics provide a short description for essays of each quality level and typical elements of such essays.

Since for the introduced task of essay feedback generation, no parallel dataset is available yet, we use the ASAP dataset as input data and evaluate the generated feedback without supervision.

5 Evaluation

We evaluate the performance of a large language model (LLM) by comparing the proposed prompting strategies on the two tasks: essay scoring and feedback generation. First, we assess the scoring performance and, then, we both automatically and manually evaluate the generated feedback in terms of the helpfulness for the student writer. We aim to study the effects of tackling essay scoring and feedback generation jointly, as well as explore how well LLMs can solve both tasks using prompting.

5.1 Essay Scoring

We compare the proposed prompt patterns, task instruction types, and in-context learning approaches, to evaluate the performance of an LLM on the essay scoring task. Also, we measure the influence of feedback generation on the scoring performance.

Approach We use the instruction-following recent LLM Mistral with 7B parameters (*Mistral-7B-Instruct-v0.2*, Jiang et al., 2023) in our experiments, generating each output with greedy decoding.[3] We found that instructing the model to generate the essay score in JSON format helps to extract the score from the generated text automatically.[4] Below, we report the number of essays that still did not receive a score (*Unscored*) and omit them from the performance calculation.

Baselines As a baseline, we report the performance AES-Prompt (Tao et al., 2022), which is, to our knowledge, the best-performing AES approach that is not fully fine-tuned on the ASAP dataset. As an upper bound, we also report the performance of R^{2}BERT (Yang et al., 2020), the state-of-the-art approach fully fine-tuned on the same dataset.

Experimental Setup We automatically assess the essay scoring performance using quadratic weighted kappa (QWK), the most widely adopted metric for automatic essay scoring (Ke and Ng, 2019). Since the test set of the ASAP dataset is not publicly available, we follow Taghipour and Ng (2016) and apply their 5-fold cross-validation split. Since we perform no training, we only use the validation splits to create reasonable initial prompts and report the performance on the test splits.

Results Table 5 presents the scoring performance for each prompt pattern. We report the average QWK of all task instructions using zero-shot learning to measure the influence of the prompt pattern on the scoring performance. Using the personas "educational researcher" (ER) and "teacher's assistant" (TA) seems beneficial for essay scoring, either of which performs best on all but one essay set, and ER best on average (mean QWK of .509).

To evaluate the influence of the task instruction type, Table 6 shows the performance of the best-

[2] For the few-shot variation, the described example selection process led to 3, 2, 4, 5, 8, 6, 4 and 2 examples for the essay sets 1 to 8 respectively. The differences are due to the variation in essay and reasoning length per essay set.

[3] Initial experiments on essay scoring with Llama-2 (*7b-chat-hf* and *13b-chat-hf*, Touvron et al., 2023) led to lower performance, which halted further testing with Llama-2.

[4] If the score was not generated as instructed, we re-prompted the model to extract the score from its prior response. This was effective when a score was in the initial answer.

Task Instruction Type	Essay Set									Unscored
	1	2	3	4	5	6	7	8	Mean	
Scoring	.448	.585	.479	.596	.557	.649	.438	.481	.529	1
Scoring→Feedback	.510	**.615**	.439	.530	.489	.621	.449	.481	.517	1
Feedback→Scoring	.388	.561	.484	.600	**.622**	.630	.385	**.545**	.527	16
Scoring→Feedback_CoT	.538	.595	.422	.494	.530	.635	.458	.477	.519	19
Feedback_dCoT→Scoring	**.546**	.564	.424	.558	.581	.628	.477	.489	**.533**	37
Scoring→Explanation	.466	.580	.472	.565	.541	.639	.420	.417	.513	0
Explanation→Scoring	.470	.553	**.488**	.636	.571	**.675**	.384	.484	**.533**	2

Table 6: Essay scoring results: QWK for the best approach variation per task instruction type in the zero-shot setting. We report the performance per essay set and the average over essay sets. The best results per column are bold.

performing approach variations per task instruction type. We report the combination of prompt pattern and task instruction that performed best on the validation set using zero-shot learning. The results suggest that instructing the LLM to first follow task-specific steps to analyze and give feedback (*Feedback_dCoT→Scoring*) as well as to first generate an explanation for the essay score (*Explanation→Scoring*) particularly help with essay scoring. These two achieve the highest mean QWK (.553). In general, the variations that generate some form of feedback first perform better than their counterparts that perform scoring first.

Finally, we study the influence of in-context learning on the instruction type *Scoring→Feedback* using the prompt pattern and task instruction that performs best on the validation split for a fair comparison to the baselines (Table 7). The results indicate that giving examples of scored essays aid essay scoring. One-shot learning outperforms few-shot learning, but the effect is rather small. Our prompting approaches perform rather competitively to the strong baseline AES-Prompt (Tao et al., 2022).

5.2 Essay Feedback Generation

As with essay scoring, we evaluate the generated feedback by comparing the prompt patterns, task instruction types, and in-context learning approaches. Our goal is to explore how well LLMs perform at generating helpful essay feedback and whether essay scoring can benefit the feedback generation.

Approach We continue using the large language model Mistral (*Mistral-7B-Instruct-v0.2*, Jiang et al., 2023) for the essay feedback generation task since it performed well at the essay scoring task.

Automatic Evaluation Using LLMs to assess the quality of generated texts has been shown to be consistent with human expert annotations for some free-text generation tasks (Chiang and Lee, 2023). Since there are no existing automatic metrics for assessing the quality of generated essay feedback, we follow previous work and use Mistral itself as well as Llama-2 (*Llama-2-13b-chat-hf*, Touvron et al., 2023) for the automatic part of our feedback evaluation. We instruct them to assign an overall helpfulness scores between 1 (not helpful) and 10 (very helpful) for each generated essay feedback. The used prompt can be found in Appendix B.[5]

Context	Essay Set								
	1	2	3	4	5	6	7	8	Mean
Zero-shot	.510	.615	.439	.530	.489	.621	.449	.481	.517
One-shot	.565	**.619**	.523	.600	.606	.665	.509	.233	.540
Few-shot	.558	.586	.515	.586	.618	**.671**	.472	.297	.538
AES-Pro.	**.682**	.544	**.590**	**.672**	**.701**	.622	**.683**	**.620**	**.639**
R^{2}BERT	.817	.719	.698	.845	.841	.847	.839	.744	.794

Table 7: Essay scoring results: QWK per in-context learning approach for *Scoring→Feedback* using the best-performing prompt pattern and task instruction. The baseline *AES-Prompt* (Tao et al., 2022) also has one shot. *R^{2}BERT* (Yang et al., 2020) is fully fine-tuned.

Our evaluation focuses on helpfulness, which we deem to be the most important quality dimension for essay feedback. We anticipate that other quality aspects, such as faithfulness, are implicitly covered since irrelevant or incorrect feedback would not be helpful for the student author.

Automatic Results Table 8 presents the assigned helpfulness scores for each prompt pattern, averaged over task instructions using zero-shot learning. Both LLMs deemed the feedback generated by a persona pattern to be most helpful, on average: the top helpfulness score is achieved by ER for Mistral (8.26) and CWM for Llama-2 (7.48).

To evaluate the influence of the task instruction

[5] We also experimented with relative comparisons of feedback for automatic helpfulness assessment. However, the correlation to our manual helpfulness annotations was low.

Prompt Pattern	Mistral	Llama-2
Base	7.78 ±0.53	6.88 ±0.18
Teacher's assistant (TA)	7.90 ±0.39	6.84 ±0.19
Educational researcher (ER)	**8.26** ±0.23	6.87 ±0.18
Creative writing mentor (CWM)	7.83 ±0.47	**7.48** ±0.85

Table 8: Automatic feedback generation results: Average helpfulness scores predicted by Mistral and Llama-2 for each prompt pattern over all task instructions using zero-shot learning. The best result per column is bold.

Task Instruction Type	Mistral	Llama-2
Feedback	**8.96** ±.25	**7.31** ±.19
Scoring→Feedback	8.04 ±.44	7.15 ±.45
Feedback→Scoring	8.27 ±.38	7.27 ±.50
Scoring→Feedback_CoT	7.30 ±.63	6.72 ±.41
Feedback_dCoT→Scoring	8.53 ±.66	7.28 ±.55
Scoring→Explanation	7.22 ±.45	6.68 ±.40
Explanation→Scoring	7.27 ±.63	6.75 ±.36

Table 9: Automatic feedback generation results: Average helpfulness scores predicted by Mistral or Llama-2 for each task instruction type over all task instructions and prompt patterns using zero-shot learning.

In-Context Learning	Mistral	Llama-2
Zero-shot learning	8.04 ±.44	7.15 ±.45
One-shot learning	8.39 ±.54	7.28 ±.47
Few-shot learning	**8.42** ±.56	**7.30** ±.46

Table 10: Automatic feedback generation results: Average helpfulness scores predicted by Mistral or Llama-2 per in-context learning approach for *Scoring→Feedback* over all prompt patterns and task instructions.

Task Instruction Type	S1	S2	S3	S4	S5
Feedback	**5.88**	**5.71**	**6.04**	**5.75**	**6.08**
Feedback→Scoring	5.17	5.04	5.46	5.21	5.08
Feedback_dCoT→Scoring	5.50	4.92	5.29	4.83	5.00

Table 11: Manual feedback generation results: Average scores assigned by the annotators for each approach for statements S1–S5 on a 7-point Likert scale (7 is best).

type, Table 9 shows the results per type, averaged over prompt patterns and task instructions using zero-shot learning. Both evaluation models gave the highest average scores to performing feedback generation only (*Feedback*). For the other task instruction types, the variations that generate some form of feedback first seem more helpful than their counterparts that perform scoring first.

Finally, we study the influence of each in-context learning approach on the task instruction type *Scoring→Feedback* on average over the prompt patterns and task instructions (Table 10). The results suggest that the reasoning presented in the provided in-context examples positively impacts the feedback helpfulness. Although the effect is small, more examples help more.

Manual Evaluation The proposed automatic evaluation only approximates the quality of the generated essay feedback. Therefore, we conducted a manual annotation study during which 12 annotators manually judged the feedback quality. All annotators have advanced English skills and are not authors of this paper. The annotators were divided into four groups that annotated the same feedback.

In particular, we randomly selected 24 essay feedback texts generated by the three task instruction types that performed best in the automatic evaluation: *Feedback*, *Feedback→Scoring*, and *Feedback_dCoT→Scoring*. Here, we used the best-performing combination of prompt pattern and task instruction. All sampled feedback texts were written for essays from one essay set only to reduce the time the annotators need to read the essay prompt. We chose essay set 4, which covers the most common ASAP task, reading comprehension.

To judge the feedback helpfulness, the annotators received the essay prompt, the student essay, and the generated feedback. Based on this, they were asked to assess to what extent the following statements apply on a 7-point Likert scale (score 1: "I strongly disagree", score 7: "I fully agree"):

S1: The feedback clearly points out mistakes that were made in the essay.

S2: The feedback explains exactly why the errors are errors.

S3: The feedback is very clear and precise so that the student can understand it.

S4: The feedback is absolutely suitable for students from 7th to 10th grade.

S5: Overall, the feedback is very helpful.

Manual Results Table 11 presents the results of the manual annotation study. For all five statements covering different helpfulness aspects, *Feedback* achieved the highest scores on average. Especially the clarity and precision (S3) as well as the overall helpfulness (S5) of *Feedback* were rated with the second-best score of 6 ("I mostly agree"). All compared task instruction types reach an average score above the neutral score of 4, indicating that all feedback is perceived as rather helpful in general. Overall, the generated essay feedback seems

Autom. Evaluation	**S1**	**S2**	**S3**	**S4**	**S5**
Mistral	0.29	0.27	0.45	0.25	**0.61**
Llama-2	–0.11	–0.11	–0.02	0.07	–0.10

Table 12: Pearson correlation of the manual annotations per statement (S1–S5) and the automatic helpfulness scores using Mistral or Llama-2. The top value is bold.

to have the most potential for improvement by better explaining why an error is erroneous (S2) and being more suitable for students (S4). The inter-annotator agreement in terms of Krippendorff's α on average over the four groups is 0.44.

To evaluate the reliability of our automatic helpfulness evaluation, we show the correlation between manual and automatic helpfulness scores in Table 12. The highest correlation value (0.61) was measured between the manually annotated overall helpfulness (S5) and the automatic helpfulness scores predicted by Mistral. This indicates that using Mistral can be useful for automatically evaluating feedback helpfulness. The helpfulness scores generated by Llama-2 do not correlate with the manual annotation for any statement.

6 Conclusion

Despite the strong text generation abilities of recent LLMs in various tasks, their effectiveness in generating essay feedback that helps student writers improve their essays has remained unclear until now. Also, generating textual feedback that addresses the entire essay has previously only been tackled using one prompting strategy in a zero-shot learning setting. With this work, we go beyond existing work by comparing different LLM prompting strategies for essay feedback generation. We propose tackling essay feedback generation and automated essay scoring (AES) jointly to study whether AES can benefit feedback generation and vice versa. Our experiments suggest that AES can be solved competitively by prompting LLMs, benefitting from tackling feedback generation first. The generated feedback is deemed helpful for students by our automatic and manual evaluation. However, the impact of scoring on the feedback helpfulness remains low ultimately.

7 Limitations

Aside from the still-improvable performance of the presented prompting approaches to automated essay scoring and feedback generation, we see two notable limitations of our work: the dependence of our feedback approaches on additional data and the pending utilization of the generated essay feedback for real-world essay writing support.

First, we point out that our feedback approaches rely on the availability of a detailed rubric, that is, guidelines including a short description for essays of each quality level, typical elements of such, and textual reasoning as to why example essays received a specific score. Such information might not always be available, which could reduce the transferability of our results to other essay datasets.

Second, while our evaluation suggests that the generated essay feedback is helpful for student writers, it remains unclear whether the student writers also perceive it as such. We encourage future work to utilize our approaches for real-world essay writing support and make it available to students. Feedback from students on such a tool would be useful to guide research on essay feedback generation.

Acknowledgments

We would like to thank the participants of our study and the anonymous reviewers for the valuable feedback and their time. This work was partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) within the project ArgSchool, project number 453073654.

References

Aniket Ajit Tambe and Manasi Kulkarni. 2022. Automated essay scoring system with grammar score analysis. In *2022 Smart Technologies, Communication and Robotics (STCR)*, pages 1–7.

Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. 2016. Automatic text scoring using neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 715–725, Berlin, Germany. Association for Computational Linguistics.

Nikolay Babakov, Maria Lysyuk, Alexander Shvets, Lilya Kazakova, and Alexander Panchenko. 2023. Error syntax aware augmentation of feedback comment generation dataset. In *Proceedings of the 16th International Natural Language Generation Conference: Generation Challenges*, pages 37–44, Prague, Czechia. Association for Computational Linguistics.

Robert L Bangert-Drowns, Marlene M Hurley, and Barbara Wilkinson. 2004. The effects of school-based writing-to-learn interventions on academic achievement: A meta-analysis. *Review of Educational Research*, 74(1):29–58.

Shabnam Behzad, Amir Zeldes, and Nathan Schneider. 2023. Sentence-level feedback generation for English language learners: Does data augmentation help? In *Proceedings of the 16th International Natural Language Generation Conference: Generation Challenges*, pages 53–59, Prague, Czechia. Association for Computational Linguistics.

Daniel Blanchard, Joel Tetreault, Derrick Higgins, Aoife Cahill, and Martin Chodorow. 2013. Toefl11: A corpus of non-native english. *ETS Research Report Series*, 2013(2):i–15.

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.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.

Mădălina Cozma, Andrei Butnaru, and Radu Tudor Ionescu. 2018. Automated essay scoring with string kernels and word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 503–509, Melbourne, Australia. 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.

Thomas Eckes. 2015. *Introduction to Many-Facet Rasch Measurement*. Peter Lang Verlag, Berlin, Deutschland.

Youmna Farag, Helen Yannakoudakis, and Ted Briscoe. 2018. Neural automated essay scoring and coherence modeling for adversarially crafted input. 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 263–271, New Orleans, Louisiana. Association for Computational Linguistics.

Tri Febriani. 2022. "Writing is challenging": factors contributing to undergraduate students' difficulties in writing English essays. *Erudita: Journal of English Language Teaching*, 2:83–93.

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.

Ben Hamner, Jaison Morgan, Iynnvandev, Mark Shermis, and Tom Vander Ark. 2012. The hewlett foundation: Automated essay scoring.

Jieun Han, Haneul Yoo, Junho Myung, Minsun Kim, Hyunseung Lim, Yoonsu Kim, Tak Yeon Lee, Hwajung Hong, Juho Kim, So-Yeon Ahn, and Alice Oh. 2023. Fabric: Automated scoring and feedback generation for essays.

Kazuaki Hanawa, Ryo Nagata, and Kentaro Inui. 2021. Exploring methods for generating feedback comments for writing learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9719–9730, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

John Hattie and Helen Timperley. 2007. The power of feedback. *Review of Educational Research*, 77(1):81–112.

Mohamed A. Hussein, Hesham A. Hassan, and Mohammad Nassef. 2020. A trait-based deep learning automated essay scoring system with adaptive feedback. *International Journal of Advanced Computer Science and Applications*, 11(5).

Mana Ihori, Hiroshi Sato, Tomohiro Tanaka, and Ryo Masumura. 2023. Retrieval, masking, and generation: Feedback comment generation using masked comment examples. In *Proceedings of the 16th International Natural Language Generation Conference: Generation Challenges*, pages 60–67, Prague, Czechia. Association for Computational Linguistics.

Kenji Imamura, Kuniko Saito, Kugatsu Sadamitsu, and Hitoshi Nishikawa. 2012. Grammar error correction using pseudo-error sentences and domain adaptation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 388–392, Jeju Island, Korea. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Kunitaka Jimichi, Kotaro Funakoshi, and Manabu Okumura. 2023. Feedback comment generation using predicted grammatical terms. In *Proceedings of the 16th International Natural Language Generation Conference: Generation Challenges*, pages 79–83, Prague, Czechia. Association for Computational Linguistics.

You-Jin Jong, Yong-Jin Kim, and Ok-Chol Ri. 2023. Review of feedback in automated essay scoring.

Noor Lide Abu Kassim. 2011. Judging behaviour and rater errors: an application of the many-facet rasch model. *GEMA Online™ Journal of Language Studies*, 179.

Zixuan Ke and Vincent Ng. 2019. Automated essay scoring: A survey of the state of the art. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 6300–6308. International Joint Conferences on Artificial Intelligence Organization.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*.

Shota Koyama, Hiroya Takamura, and Naoaki Okazaki. 2023. The Tokyo tech and AIST system at the GenChal 2022 shared task on feedback comment generation. In *Proceedings of the 16th International Natural Language Generation Conference: Generation Challenges*, pages 74–78, Prague, Czechia. Association for Computational Linguistics.

Vivekanandan Kumar and David Boulanger. 2020. Explainable automated essay scoring: Deep learning really has pedagogical value. *Frontiers in Education*, 5.

Alina Leidinger, Robert van Rooij, and Ekaterina Shutova. 2023. The language of prompting: What linguistic properties make a prompt successful? In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9210–9232, Singapore. Association for Computational Linguistics.

Xia Li, Minping Chen, Jianyun Nie, Zhenxing Liu, Ziheng Feng, and Yingdan Cai. 2018. Coherence-based automated essay scoring using self-attention. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 386–397, Cham. Springer International Publishing.

Ou Lydia Liu, Lois Frankel, and Katrina Crotts Roohr. 2014. Assessing critical thinking in higher education: Current state and directions for next-generation assessment. *ETS Research Report Series*, 2014(1):1–23.

Sandeep Mathias and Pushpak Bhattacharyya. 2020. Can neural networks automatically score essay traits? In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 85–91, Seattle, WA, USA → Online. Association for Computational Linguistics.

Jennifer Meyer, Thorben Jansen, Ronja Schiller, Lucas W. Liebenow, Marlene Steinbach, Andrea Horbach, and Johanna Fleckenstein. 2024. Using llms to bring evidence-based feedback into the classroom: Ai-generated feedback increases secondary students' text revision, motivation, and positive emotions. *Computers and Education: Artificial Intelligence*, 6:100199.

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.

Ryo Nagata. 2019. Toward a task of feedback comment generation for writing learning. 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 3206–3215, Hong Kong, China. Association for Computational Linguistics.

OpenAI. 2023. ChatGPT (GPT version: 3.5). Large language model.

John Peloghitis. 2017. Difficulties and strategies in argumentative writing: A qualitative analysis. In *Transformation in language education*, Tokyo. JALT.

Isaac Persing, Alan Davis, and Vincent Ng. 2010. Modeling organization in student essays. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 229–239, Cambridge, MA. Association for Computational Linguistics.

Zahra Rahimi, Diane Litman, Elaine Wang, and Richard Correnti. 2015. Incorporating coherence of topics as a criterion in automatic response-to-text assessment of the organization of writing. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 20–30, Denver, Colorado. Association for Computational Linguistics.

Dadi Ramesh and Suresh Kumar Sanampudi. 2022. An automated essay scoring systems: a systematic literature review. *Artificial Intelligence Review*, 55(3):2495–2527.

Jessica Riddell. 2015. Performance, feedback, and revision: Metacognitive approaches to undergraduate essay writing. *Collected Essays on Learning and Teaching*, 8:79.

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.

Rebecca Schendel and Andrew Tolmie. 2016. Beyond translation: adapting a performance-task-based assessment of critical thinking ability for use in rwanda. *Assessment & Evaluation in Higher Education*, 42(5):673–689.

Valerie J. Shute. 2008. Focus on formative feedback. *Review of Educational Research*, 78(1):153–189.

Yixiao Song, Kalpesh Krishna, Rajesh Bhatt, Kevin Gimpel, and Mohit Iyyer. 2023. Gee! grammar error explanation with large language models.

Maja Stahl and Henning Wachsmuth. 2023. Identifying feedback types to augment feedback comment generation. In *Proceedings of the 16th International Natural Language Generation Conference: Generation Challenges*, pages 31–36, Prague, Czechia. Association for Computational Linguistics.

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.

Qiuyu Tao, Jiang Zhong, and Rongzhen Li. 2022. Aesprompt: Self-supervised constraints for automated essay scoring with prompt tuning. In *The 34th International Conference on Software Engineering and Knowledge Engineering, SEKE 2022, KSIR Virtual Conference Center, USA, July 1 - July 10, 2022*, pages 335–340. KSI Research Inc.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Masaki Uto. 2021. A review of deep-neural automated essay scoring models. *Behaviormetrika*, 48(2):459–484.

Masaki Uto, Yikuan Xie, and Maomi Ueno. 2020. Neural automated essay scoring incorporating handcrafted features. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6077–6088, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Sowmya Vajjala. 2018. Automated assessment of non-native learner essays: Investigating the role of linguistic features. *International Journal of Artificial Intelligence in Education*, 28(1):79–105.

Fabienne M. Van der Kleij, Remco C. W. Feskens, and Theo J. H. M. Eggen. 2015. Effects of feedback in a computer-based learning environment on students' learning outcomes: A meta-analysis. *Review of Educational Research*, 85(4):475–511.

Cong Wang, Zhiwei Jiang, Yafeng Yin, Zifeng Cheng, Shiping Ge, and Qing Gu. 2023. Aggregating multiple heuristic signals as supervision for unsupervised automated essay scoring. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13999–14013, Toronto, Canada. Association for Computational Linguistics.

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*.

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C. Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt.

Jiayi Xie, Kaiwei Cai, Li Kong, Junsheng Zhou, and Weiguang Qu. 2022. Automated essay scoring via pairwise contrastive regression. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2724–2733, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Ruosong Yang, Jiannong Cao, Zhiyuan Wen, Youzheng Wu, and Xiaodong He. 2020. Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1560–1569, Online. Association for Computational Linguistics.

Xi Ye and Greg Durrett. 2022. The unreliability of explanations in few-shot prompting for textual reasoning. In *Advances in Neural Information Processing Systems*.

A Task Instructions

We present all used task instructions in the following list. This includes all paraphrases per task instruction type.

- **Scoring:** *(1)* Given this essay that was written for the given prompt, grade the essay using those ranges: {*scoring_range*}.
  (2) Review the provided essay in response to the given prompt. Assess its quality and assign a grade according to the following criteria: {*scoring_range*}.

(3) Examine the essay written in response to the specified prompt. Utilize the following grading ranges to evaluate and score the essay: {*scoring_range*}.
(4) Analyze the submitted essay that corresponds to the given prompt. Apply these grading standards to determine its score: {*scoring_range*}.

- **Feedback:** *(1)* Analyze the given essay using the following rubric: {*rubric*}. Provide comprehensive feedback for the student that helps them to achieve better grades in the future.
  (2) Please evaluate the essay in accordance with the criteria outlined in: {*rubric*}. Offer detailed and constructive feedback to assist the student in improving their writing skills for future assignments.
  (3) Utilize the provided rubric ({*rubric*}) to assess the essay. Your feedback should be thorough, focusing on areas of strength and suggesting improvements to help the student enhance their academic writing.
  (4) Conduct an assessment of the submitted essay using this specific rubric: {*rubric*}. Your feedback should be insightful and supportive, guiding the student towards achieving higher grades in their future essays.

- **Scoring→Feedback:** *(1)* Grade the given essay using the following rubric: {*rubric*}. Use those score ranges: {*scoring_range*}. Provide comprehensive feedback for the student that helps them to achieve better grades in the future.
  (2) Please evaluate the essay in accordance with the criteria outlined in: {*rubric*}. Assign a grade based on these standards: {*scoring_range*}. Offer detailed and constructive feedback to assist the student in improving their writing skills for future assignments.
  (3) Utilize the provided rubric ({*rubric*}) to assess the essay. Grade it according to these parameters: {*scoring_range*}. Your feedback should be thorough, focusing on areas of strength and suggesting improvements to help the student enhance their academic writing.
  (4) Conduct an assessment of the submitted essay using this specific rubric: {*rubric*}. Apply the grading criteria as per these guidelines: {*scoring_range*}. Your feedback should be insightful and supportive, guiding the student towards achieving higher grades in their future essays.

- **Feedback→Scoring:** *(1)* Analyse the given essay using the following rubric: {*rubric*}. Use those score ranges: {*scoring_range*}. To do this, first provide comprehensive feedback for the student that helps them to achieve better grades in the future. Then give the final score.
  (2) Begin by carefully reviewing the submitted essay in light of the criteria outlined in {*rubric*}. After your thorough analysis, offer detailed and constructive feedback aimed at guiding the student towards academic improvement. Conclude your review by assigning a score to the essay, adhering to the guidelines specified in {*scoring_range*}.
  (3) First, evaluate the essay against the criteria mentioned in {*rubric*}. Your evaluation should include specific, actionable suggestions for the student to enhance their writing skills and essay quality. Following your comprehensive feedback, assign a score to the essay based on the scale provided in {*scoring_range*}.
  (4) Commence your assessment by applying the criteria from {*rubric*} to the essay. Focus on delivering in-depth feedback that is both informative and beneficial for the student's future academic endeavors. After providing this feedback, conclude by scoring the essay as per the range defined in {*scoring_range*}.

- **Scoring→Feedback_CoT:** *(1)* Analyse the given essay using the following rubric and give helpful feedback to the student: {*rubric*}. Use those score ranges: {*scoring_range*}. Let's think step by step. Make sure to output the score only at the end.
  (2) Please evaluate the provided essay according to this specific rubric: {*rubric*}. Scores should be assigned based on these criteria: {*scoring_range*}. Proceed methodically through each step. Conclude your analysis by presenting the final score.
  (3) Conduct a thorough assessment of the essay using the rubric below: {*rubric*}. Adhere to the following scoring guidelines: {*scoring_range*}. Break down your analysis into clear steps. Ensure the final score is given at the end of your evaluation.

(4) Examine the student's essay in detail, utilizing the rubric provided: {*rubric*}. Apply these scoring ranges for evaluation: {*scoring_range*}. Tackle the analysis in a step-by-step manner. The score should be presented at the conclusion of your feedback.

- **Feedback_dCoT→Scoring:** *(1)* Analyze the given essay using the following rubric and give helpful feedback to the student: {*rubric*}. Use those score ranges: {*scoring_range*}. Let's think step by step. First, analyze the quality of the essay in terms of the given rubric. Then, give feedback to the student that explains their mistakes and errors and additionally gives them tips to avoid them in the future. As a final step, output the score at the end.
(2) Begin by evaluating the essay based on the criteria outlined in the rubric: {*rubric*}. Consider the scoring guidelines provided: {*scoring_range*}. First, conduct a thorough analysis of the essay according to the rubric standards. Next, provide constructive feedback to the student, highlighting areas for improvement and suggesting strategies to enhance their writing skills. Conclude with a summary of the essay's strengths and weaknesses. Finally, present the essay's score at the end of your analysis.
(3) Follow these steps to assess the student's essay: First, reference the provided rubric: {*rubric*}, and apply it to evaluate the essay. Use the scoring ranges given: {*scoring_range*} for accurate assessment. Provide detailed feedback to the student, pinpointing specific areas of the essay that align or deviate from the rubric, along with advice for future improvement. Your feedback should be clear, constructive, and actionable. After your comprehensive review, conclude by outputting the final score, ensuring this is done only at the very end.
(4) To evaluate the student's essay, proceed as follows: Start with the provided rubric: {*rubric*}, to assess the essay's attributes. Adhere to the scoring guidelines: {*scoring_range*} for consistency. Your analysis should first focus on how well the essay meets the criteria in the rubric. Then, craft feedback for the student that is both informative and helpful, addressing any shortcomings and providing practical advice for future essays. The feedback should be encouraging yet honest. Conclude your evaluation by scoring the essay, presented at the conclusion of your feedback.

- **Scoring→Explanation:** *(1)* Grade the given essay using the following rubric: {*rubric*}. Use those score ranges: {*scoring_range*}. Provide an explanation for your score as well.
(2) Please assess the submitted essay according to the criteria outlined in this rubric: {*rubric*}. Scores should be allocated based on these guidelines: {*scoring_range*}. Additionally, include a detailed rationale for the score you assign.
(3) Evaluate the provided essay by referring to the standards specified here: {*rubric*}. Utilize the following scoring range for your evaluation: {*scoring_range*}. Also, furnish a comprehensive justification for the grade you determine.
(4) Rate the essay in front of you using these evaluation criteria: {*rubric*}. Your scoring should align with these parameters: {*scoring_range*}. Please also give a thorough explanation to support the score you decide upon.

- **Explanation→Scoring:** *(1)* Analyse the given essay using the following rubric: {*rubric*}. To do this, first explain using the scoring rubric why you chose the score. After you analysed the essay, give a final grade.
(2) Utilize the provided scoring rubric ({*rubric*}) to evaluate the essay. Begin by detailing the reasons for your assigned score based on the rubric's criteria. Conclude by summarizing your analysis with a final grade.
(3) Apply the scoring guidelines from {*rubric*} to assess the essay. Start by discussing how the essay meets or falls short of each criterion in the rubric. Finalize your assessment with a grade.
(4) Use the scoring rubric ({*rubric*}) as a basis to critically analyze the essay. Explain how the essay aligns with each aspect of the rubric, justifying your evaluation. Conclude with a definitive grade.

B Automatic Helpfulness Assessment

We used the following prompt for our automatic helpfulness evaluation:

"You are given an essay and feedback from a teacher for this essay. Your task is to evaluate the helpfulness of the feedback.

# Task: Evaluate the helpfulness of the feedback. Helpful feedback should explain what the errors are, why they are errors, and how to fix them. Give a score between 1 and 10, where 1 means the feedback is not helpful at all, and 10 means the feedback is very helpful.

Provide the output in the following output: {format_instructions}"

C Generated Essay Feedback

Table 13 compares the generated essay feedback for the same essay using three different task instruction types, compared in the manual evaluation.

Task Instruction Type	**Generated Essay Feedback**
Feedback	Based on the provided essay prompt and the given text, it appears that the student has not fully understood the story "Winter Hibiscus" by Minfong Ho. The student's response, "Saeng would return to her homeland," is not an accurate reflection of the story's conclusion. The author concludes the story with Saeng's determination to retake the driver's test in the spring when the hibiscus is budding and the geese have returned. This decision comes after Saeng's emotional experience in the greenhouse, where she is surrounded by familiar plants and memories, and is struck by a wave of loss for her past life in Vietnam. The author's use of the imagery of the geese returning and the hibiscus budding signifies the cyclical nature of life and the passing of seasons. Saeng's decision to retake the driver's test in the spring shows her resilience and determination to move forward in her new life in the United States while holding on to her past. To improve future essays, the student could focus on closely reading and understanding the text, taking notes, and identifying key themes and symbols. It may also be helpful for the student to practice summarizing and paraphrasing the text in their own words to demonstrate a deeper understanding.
Feedback→Scoring	The essay response is very brief and does not fully address the demands of the question. The statement 'Saeng would return to her homeland' is not explicitly stated in the text, and the essay does not provide any evidence or explanation to support this assertion. To improve, consider reading the text more closely and taking notes to help you understand the main ideas and themes. Use these notes to develop a more detailed and well-supported response that addresses the demands of the question.
Feedback_dCoT→Scoring	Your essay does not fully address the demands of the question. In order to earn a higher score, you should explain why Saeng would return to her homeland based on the details and events in the story. Consider how the hibiscus plant and the memories it evokes for Saeng might be connected to her desire to go back to her homeland. Additionally, you could discuss how the experiences and emotions Saeng encounters in the greenhouse contribute to her decision. Remember to use specific details and examples from the text to support your ideas.

Table 13: Generated essay feedback using the three task instruction types compared in the manual annotation study. The feedback was generated for the same student essay from essay set 4 of the ASAP dataset (Hamner et al., 2012).

Towards Fine-Grained Pedagogical Control over English Grammar Complexity in Educational Text Generation

Dominik Glandorf
University of Tübingen
Yale University
dominik.glandorf@student.uni-tuebingen.de

Detmar Meurers
Leibniz-Institut für Wissensmedien (IWM)
detmar.meurers@uni-tuebingen.de

Abstract

Teaching foreign languages and fostering language awareness in subject matter teaching requires a profound knowledge of grammar structures. Yet, while Large Language Models can act as tutors, it is unclear how effectively they can control grammar in generated text and adapt to learner needs. In this study, we investigate the ability of these models to exemplify pedagogically relevant grammar patterns, detect instances of grammar in a given text, and constrain text generation to grammar characteristic of a proficiency level. Concretely, we (1) evaluate the ability of GPT3.5 and GPT4 to generate example sentences for the standard English Grammar Profile CEFR taxonomy using few-shot in-context learning, (2) train BERT-based detectors with these generated examples of grammatical patterns, and (3) control the grammatical complexity of text generated by the open Mistral model by ranking sentence candidates with these detectors. We show that the grammar pattern instantiation quality is accurate but too homogeneous, and our classifiers successfully detect these patterns. A GPT-generated dataset of almost 1 million positive and negative examples for the English Grammar Profile is released with this work. With our method, Mistral's output significantly increases the number of characteristic grammar constructions on the desired level, outperforming GPT4. This showcases how language domain knowledge can enhance Large Language Models for specific education needs, facilitating their effective use for intelligent tutor development and AI-generated materials. Code, models, and data are available at https://github.com/dominikglandorf/LLM-grammar.

1 Introduction

The arrival and accessibility of well-performing Large Language Models (LLMs) created a flood of applications in personalized education for tutoring and material creation (Kasneci et al., 2023). Despite their ability to follow instructions, it is underexplored to what extent prompting can systematically affect the linguistic properties of the generated output to satisfy educational needs. If LLM-generated text was finely adjustable regarding the grammatical constructs used, personalized and engaging learning materials could systematically support learners' language development by exposing them to the optimal linguistic complexity (Mart, 2013). This control would enable a stronger connection to input-oriented theories of language acquisition.

Due to their data-driven nature, LLMs' grammatical knowledge has to be empirically examined. On the one hand, they have been successfully used for text simplification and grammar construction detection (Jeblick et al., 2023; Weissweiler et al., 2022). On the other hand, transformer models still benefit from explicit syntactic information during training (Hu et al., 2020). Because of missing labeled training data and systematic evaluations, it is uncertain to what extent neural text generation can be controlled for the presence of a comprehensive set of pedagogically relevant and teachable grammatical constructions.

This work pursues the questions of how well LLMs can create valid examples for grammar constructs (RQ1), how well BERT sentence embeddings represent these grammar constructs (RQ2), and how well text generation can be controlled for these constructs (RQ3). We build on an empirically established and validated taxonomy of English grammar, the English Grammar Profile (EGP) (O'Keeffe and Mark, 2017), precisely characterizing the development of English across the proficiency spectrum with 1,222 grammar patterns. We first evaluate how well GPT3.5 and GPT4 can generate positive and negative instances on a subset of the EGP (RQ1). We then alleviate the lack of examples by automatically creating 946K labeled example sentences for all entries of the EGP, which we make available to the public. This unique dataset serves to fine-tune and evaluate BERT-based classification models on detecting examples of the EGP's grammar patterns in sentences (RQ2). Using these models, a grammar-controlled text generation approach to strategically decoding an open pre-trained LLM, Mistral-7B, provides a proof of concept with 600 generated texts (RQ3). To generate them, we sample multiple candidate sentences at inference time and rank them by the grammar patterns detected by the classifiers.

We show that the accuracy of generated instances of grammar patterns is 87.1% with GPT3.5 (92.9% with GPT4), and the classifiers distinguish the positive from negative examples in our generated dataset with an average accuracy of 95.1%. The grammar-controlled text generation approach at least doubles the grammatical constructions on each level of the standard Common European Framework of Reference for Languages (CEFR)({Council of Europe}, 2020).

Going beyond the specific task, our work highlights how explicit domain knowledge relevant to language learning and broader language-sensitive educational contexts can be fused with the versatility of LLMs. It is a step towards better control over a powerful tool compared to pure prompting. The approach can readily be extended to other pedagogically desirable attributes of LLM-based tutors and educational material.

2 Related Work

2.1 Grammatical complexity in education

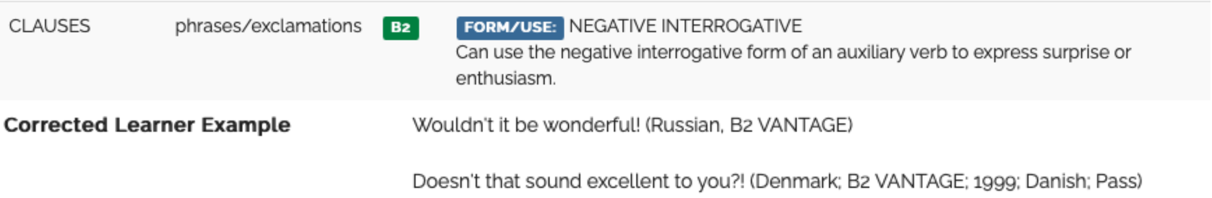
Krashen's Input Hypothesis about language learning features the idea that input is an essential driver of language development if understandable to a learner but one step beyond their language level (Krashen, 1992). Although criticized for the vagueness of the theory's predictions, the role of input is broadly accepted in the literature (Lichtman and VanPatten, 2021; Loewen, 2021; Ellis, 2002). Learners benefit from language input adapted to their proficiency level. This assumption manifests itself in *graded readers*, such as simplified literature for learners. Not only do they adapt lexical features but also grammatical complexity (Zakaria et al., 2023). Berendes et al. (2018) systematically analyzed textbooks and highlighted the need to pay more attention to language complexity in subject-matter teaching regarding learner appropriateness. Indeed, research on language-sensitive education in science and other subjects stresses that learning difficulties often arise due to factors such as the syntactic complexity of the language used (Wellington and Osborne, 2001). The success of graded readers and these shortcomings underline the importance of controlling grammar in learning materials for effective language development and the potential impact of automated control.

O'Keeffe and Mark (2017) compiled and published the English Grammar Profile based on the systematic analysis of learner data from language proficiency exams. The EGP includes 1,222 grammar constructs that learners use on different levels, categorized by the standard CEFR level, from A1 (beginner) to C2 (native). They are organized into 19 categories (e.g., adverbs) and can be of type *FORM*, *FORM/USE*, or *USE*. FORM means constructs that can be described lexically and syntactically, whereas USE refers to a semantic function of a linguistic form. The EGP includes a brief description in the form of a can-do statement and one to five authentic learner examples for each structure, as illustrated in Figure 1.

Research on fostering adaptive language learning has started to use developmentally proximal input, though it typically does so by selecting from existing materials (Chen et al., 2022). The EGP's instance-based characteristics of grammatical development allow for fine-grained adaptivity in language teaching because each construct is teachable (and indeed, many are explicitly specified as part of school curricula), which contrasts with the typical aggregate measures and ratios used in linguistic complexity research as part of the Complexity, Accuracy, and Fluency triad (Housen et al., 2012). Thus, the EGP can be a milestone in measuring the grammar complexity of learner input, which is especially valuable when generating material for learners in earlier stages of development, for which little authentic language material exists. However, no large-scale corpus annotated with the EGP constructions is publicly available, yielding the need for our novel dataset.

2.2 Grammar-related tasks in natural language processing

Recent LLMs are performant on high-level grammar-related tasks such as essay complexity scoring (Yancey et al., 2023) and text simplification (Jeblick et al., 2023), suggesting a general grasp of grammatical structures. Low-level tasks include grammar annotation, for example with a pre-trained



CLAUSES	phrases/exclamations	B2	FORM/USE: NEGATIVE INTERROGATIVE Can use the negative interrogative form of an auxiliary verb to express surprise or enthusiasm.
Corrected Learner Example			Wouldn't it be wonderful! (Russian, B2 VANTAGE) Doesn't that sound excellent to you?! (Denmark; B2 VANTAGE; 1999; Danish; Pass)

Figure 1: An English Grammar Profile construct at level B2 with two examples

BERT model (Devlin et al., 2019). Weissweiler et al. (2022) successfully detected the presence of the comparative correlative in English with logistic regressions on BERT sentence representations. Yu et al. (2023) also argue for the potential of LLMs for linguistic annotation compared to traditional natural language processing techniques, especially for semantic features without a mapping to lexical forms. Their results for annotating acts of apologizing hint that LLMs can distinguish complex grammatical functions of words and can potentially solve tasks demanding grammatical knowledge. The only work that classified an EGP-alike set of constructions from SCoRE (Chujo et al., 2015) used BERT models to detect three constructions and was successful in increasing their likelihood in generated dialog responses via reinforcement learning (Okano et al., 2023). Unfortunately, the construction-wise reinforced models cannot be combined, making the approach challenging to scale.

Controlled text generation has developed from decoding strategies and supervised fine-tuning (Xiao et al., 2023) to prompt engineering (Koraishi, 2023) and preference optimization approaches (Rafailov et al., 2023). Apart from Okano et al. (2023), past work on syntactic constraints usually worked on parse trees or part-of-speech sequences, which are not directly mappable to curricular grammar patterns (Sun et al., 2023). Especially EGP patterns of the type *USE* are semantic and impossible to represent in this form. Advanced controlled text generation approaches are out of the scope of this work, but the resulting classifiers of this work can be incorporated into all of these approaches.

3 Method

Our approach comprises validating the EGP instantiation capabilities of state-of-the-art LLMs, training neural rule detectors on a generated large-scale grammar dataset, and using these rule detectors to rank candidates when sampling from an open text generation model. The analysis was conducted with standard Python libraries for natural language processing and deep learning on up to 16 Nvidia GeForce RTX 2080 Ti GPUs provided by the computing cluster of the University of Tübingen. The code and data are available on GitHub[1]. Seeds are provided for reproducibility.

3.1 Instantiating the English Grammar Profile

This step evaluates the possibility of automatically sourcing a high-quality labeled dataset of single grammar constructions. The English Grammar Profile is obtained from its official website[2]. Its structure is characterized in Section 2.1. The information about the learner and the uncorrected examples are removed. We prompt the OpenAI Chat Completion API[3] to generate more examples, namely positive instances of the rule and negatives that ought to have the same meaning without using the construct (i.e., a minimal pair). We evaluate two model checkpoints for comparison, `gpt-3.5-turbo-1106` and `gpt-4-0125-preview`, using in-context learning with a prompt template to describe the grammar rule and append the one to five available examples. If present, the numerical value for the lexical range is translated into *low*, *medium*, and *high*. After the list of positive examples is returned, a second prompt asks to rewrite every example as a minimal pair without using the construction. These are the exact prompts:

```
1. Learn the grammar rule "{Can-
   do statement}" ({Super
   Category}, {Sub Category}, {
   Guideword}). It is CEFR level
   {Level}. {Lexical Range}
Examples:
{Examples}
```

[1] https://github.com/dominikglandorf/LLM-grammar

[2] https://www.englishprofile.org/english-grammar-profile/egp-online

[3] https://platform.openai.com/docs/models/overview

```
Create {Batch Size} more examples
    using that rule.

2. {Previous Prompt and Response}
Rewrite each created example as a
    minimal pair that does not
   show the usage of the given
   rule.
```

Using regular expressions, the model responses are parsed based on the enumeration, cleaned from prefixes and explanations in parenthesis, and cleared from repetitions of the positive examples in the case of negative instances. The `presence_penalty` parameter that penalizes repetitions of tokens during sampling from the model was increased to 0.5 for the initial prompt to diversify the vocabulary within one response. The model temperature that makes the output more random for higher values was decreased to 0.5 for the negative prompt to favor correctness over diversity. This assumes that there is only a small number of possible modifications to make a positive example negative and therefore the sampling should favor the most likely tokens. The EGP may or may not be part of the training set of OpenAI's models. Even if this is the case, it remains unclear how well they can transfer the patterns to a wider range of topics and sentence meanings than the few included examples.

For a small-scale quality assessment (before generating the large dataset in the next step), 36 EGP patterns are randomly drawn, stratified by CEFR level and type, and the two models generate each 20 (in two batches of 10) positive and 20 negative examples each, resulting in 2,880 examples. The set of sentences is hand-coded on whether they include the intended grammar pattern or not in a blinded manner, i.e., without knowing the model or intended label. These labels serve to calculate the models' accuracy. An automatic evaluation based on the ROUGE and BLEU scores assesses how close the negative examples are to the most similar positive example. The ROUGE-1 score (ranging from 0 to 1) reflects the number of common unigrams between a text and the set of reference texts, measuring lexical similarity. The BLEU score is in the same range but focusses more on precision instead of recall and also takes longer subsequences into account. Furthermore, the average cosine similarity of embeddings with the recent `ember-v1` model[4] between all positive example sentences and between all negative sentences is calculated per EGP pattern and compared to the baseline of the renowned Brown corpus (Kučera et al., 1967). To improve the diversity of negative examples, positive examples from other EGP entries are mixed in, assuming that these do not contain the pattern.

3.2 Detecting instances of grammar patterns

This step poses the challenge of learning a binary classifier that detects the presence of a single EGP construct in a given sentence. The bidirectional transformer architecture led to a breakthrough in natural language understanding and was also used by prior work on grammar detection (Okano et al., 2023; Weissweiler et al., 2022). Due to the large number of EGP constructs, we use multi-task training. We choose BERT instead of non-neural tools due to the much lower cost of development, only requiring accurate training data. We fine-tune a pretrained instance of `bert-base-uncased` (Devlin et al., 2019) with model dimensionality 768 and 12 attention layers (110M parameters) as a shared embedding model for each of the six CEFR levels. We train for each single construction a two-layer feedforward network with a hidden dimensionality of 16 on the mean-pooled output from the shared model (12,320 extra parameters per construction). This is a compromise between optimal performance by fine-tuning an entire BERT for each construction and saving the vast amount of GPU memory that this would entail. We did not explore other model architectures because preliminary results have been satisfying.

We use `gpt-3.5-turbo-1106` to create 500 unique positive and 250 unique negative examples for *each* EGP construct in batches of 25 because the model often refused to create larger batches. This results in the large-scale dataset of 946,246 sentences we release with this work. During training, we add 250 random positive examples from other constructs labeled as negative to increase the diversity of the dataset, assuming these do not contain the rule. This leads to a total of 109K (CEFR A1) to 338K (CEFR B1) sentences to train and evaluate each of the six models. Gradients were accumulated across batches of all constructs before taking an optimizer step to balance the influence of a single construct. The batch size was 8, the learning rate for the AdamW optimizer was 0.0001, and training was stopped as soon as the validation loss increased or after a maximum of 5 epochs. We release the trained models with data.

[4] https://huggingface.co/llmrails/ember-v1

We use 5-fold cross-validation for evaluation and do not pursue systematic hyperparameter tuning due to satisfying initial experimentation results. Because of the balanced classes, accuracy is the primary evaluation metric besides precision and recall.

3.3 Controling text generation for grammar patterns

This step uses the trained classifiers to control language model output for grammar patterns. Caused by the lack of authentic text annotated with single EGP entries, the CEFR level is used as a proxy. Ideally, a text for a certain level exposes the reader to a high amount of grammar constructions on that level. Thus, the goal is to generate texts with the most EGP constructs of a given level, as indicated by the previous step's classifiers. A CEFR-labeled dataset that was compiled from online resources[5] serves as the static baseline. It contains 1,494 texts on all CEFR levels, 37,008 sentences in total.

We generate 600 texts (100 per level) with each method for comparison. As the LLM baselines, `Mistral-7B-Instruct-v0.2` (Jiang et al., 2023) and `gpt-4-0125-preview` are prompted to continue the first words of given writings with as many grammar constructs as possible on a specific CEFR level, explained with its official description ({Council of Europe}, 2020). Mistral-7B is a model with an architecture and training procedure comparable to the GPT models but with efficiency adjustments. We relied on Mistral-7B due to its appealing trade-off between model size and performance and added GPT4 as the best-performing, closed model at the time. We ran Mistral in inference mode on our cluster infrastructure on two of the GPU cores.

In our proposed ranking approach, the model, prompted in the same way as the baselines, generates five sentence candidates, and the candidate with the most grammar constructs on the desired level is chosen in the remaining generation procedure. This approach is supposed to succeed if the generated candidates show a significant variance in grammar constructions. Tyen et al. (2022) chose a similar ranking approach to generate dialog responses of a specified CEFR level but was using a classifier predicting the CEFR level of candidates instead of explicitly the presence of grammatical structures. Although possible, we did not use a smaller set of preferred EGP patterns because of the large number of constructions and the potential inefficiency of sampling. Past work has also emphasized that a single grammar rule may not be sensible in every generated sentence (Okano et al., 2023). The prompt comprised at least the first 50 characters (adding characters up to the following space) from randomly drawn texts in the CEFR dataset to set different topics of the stories:

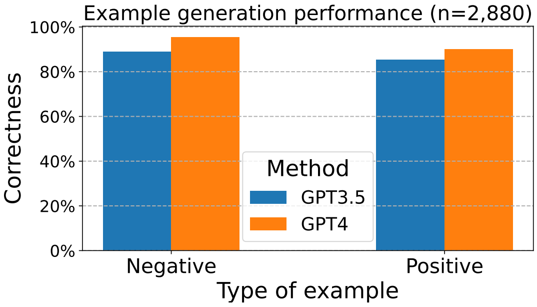


Figure 2: Ratio of correctly generated instances by model and type of example.

```
[INST] Continue the writing using
    as many grammar constructs on
   CEFR level {level} as
  possible ({level description})
  . Do not talk about the CEFR
  level.
[/INST] {story beginning}
```

We stop generation when the continuation exceeds 1,024 characters (Mistral) or 256 tokens (GPT4). The evaluation metric is the average percentage of detected constructions in the corresponding text level.

4 Results

4.1 Grammar Pattern Instantiation

4.1.1 Accuracy

Figure 2 summarizes the manually evaluated quality of the two models on generating instances of 36 sample EGP entries. On average, GPT4 got overall 92.9% of the generated instances right, while GPT3.5 scored 87.1%. This difference holds for positive and negative examples, while both models score a few percentage points worse on positive examples. This indicates that they got some rules wrong in the first place. Since all four conditions are accurate far above the random baseline of 50%, the accuracy of the LLM-generated examples is satisfying, and the next steps can build on this technique.

[5] `https://www.kaggle.com/datasets/amontgomerie/cefr-levelled-english-texts`

Table 1: ROUGE and BLEU scores of negative examples versus positive examples

Model & Parameters	ROUGE-1↑	BLEU↑
GPT3.5, temp=1	0.704	0.268
GPT3.5, temp=0.5	**0.783**	**0.368**
GPT4, temp=1	0.721	0.283

EGP#777: Can use the past perfect simple to talk about situations which changed.

+ They had expected to win the match, but their opponents played exceptionally well.
- They expected to win the match, but their opponents played exceptionally well.
+ She had thought she had everything under control, but then the unexpected happened.
- She thought she had everything under control, but then the unexpected happened.
+ We had believed we had enough time to finish the project, but unforeseen complications arose.
- We believed we had enough time to finish the project, but unforeseen complications arose.

EGP#288: Can use no article before a limited range of singular, plural and uncountable nouns when referring to things in general.

+ Dogs are my favorite animals.
- The dogs in the park are friendly.
+ Milk is good for your bones.
- The milk in the fridge is expired.
+ I don't like carrots.
- The carrots in the salad are fresh.

Figure 3: Generated positive (+) and negative (-) examples for an EGP entry with very high average ROUGE and BLUE scores (top) and one with very low scores (bottom).

4.1.2 Minimality

Table 1 shows the automatic quality assessment of the minimality of the negative examples, measured by their ROUGE and BLEU scores with respect to the positive examples. Interestingly, the temperature is more critical than the general performance of the model. Concretely, GPT3.5 with decreased temperature performs better than GPT4 with the default temperature. This hints at the importance of reducing the randomness when sampling from the language model output. Figure 3 shows generated examples for two EGP entries. These instances show that there may be rules for which minimal negative examples are easier to create. For construct #288, one could just add the article but would make the sentences potentially ungrammatical. This shows that the model also takes care of the correctness of the generated examples.

Table 2: Average cosine similarities between sentences in authentic text (Brown corpus) and the positive and negative examples generated by GPT3.5. *Random others refer to negative examples with random positive examples from other constructs.

Corpus	Similarity	Std. Dev.
Brown	**0.334**	0.002
Positive examples	0.462	0.052
Negative examples	0.451	0.045
Random others*	0.369	0.007

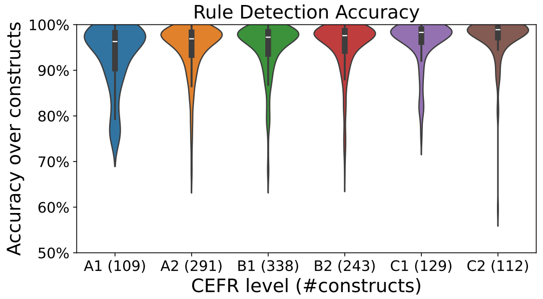


Figure 4: Accuracy distributions of the grammar classifiers across CEFR difficulty levels. Variation between cross-validation folds is negligible. The baseline is 50%. The white dash indicates the median and the pronounced black strip the interquartile range.

4.1.3 Diversity

The diversity of the generated examples, indicated by the average sentence similarity within the generated EGP patterns, is represented by Table 2.

To some extent, the similarity between the examples is expected to be higher due to the presence of the grammar pattern. Still, we observe a significantly increased cosine similarity for both positive and negative pairs compared to the Brown reference corpus. Adding positive examples from other EGP constructs increases the diversity, yielding an average cosine similarity increased by only 10% compared to the reference corpus. Overall, the evaluation confirms the capability of state-of-the-art LLMs to augment a grammar pattern dataset from a class description and a few examples with accurate positive and negative examples, only lacking diversity within the positive examples.

4.2 Grammar Pattern Detection

Figure 4 depicts the accuracies of our BERT-based models at detecting whether a given grammar construction is present in a sentence.

The average accuracy of all classifiers is 95.1%, precision is 93.3% and recall is 97.3%. The distri-

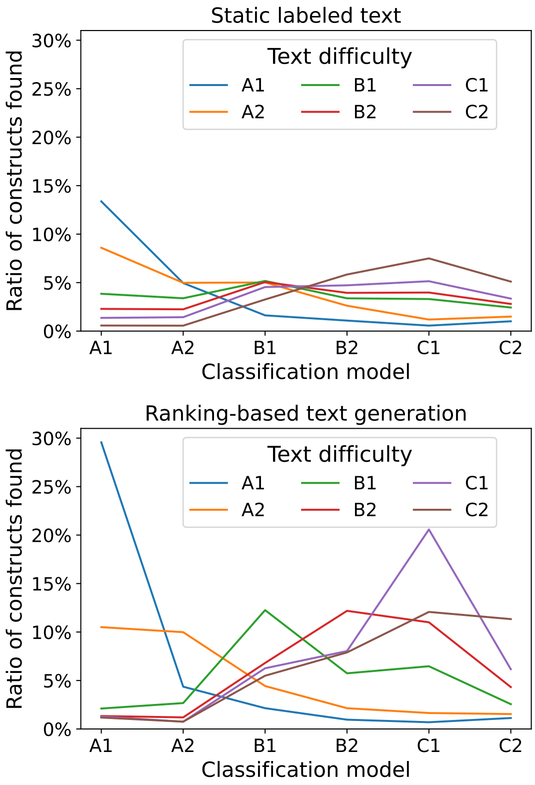


Figure 5: Number of grammatical constructions per CEFR levels for the static baseline (top) and our proposed approach (bottom). The trained classifiers from the previous step detected constructions.

butions of precisions and recalls are mean-shifted but overall very similar and are included in our GitHub repository. The average recall attains at least 91% among all CEFR levels. The lower precision may be explainable by false negatives added for diversification and the quality differences between positive and negative examples. The accuracy distribution within CEFR level A1 reveals slight problems detecting some of these constructions. This may be due to the very basic character of many A1 grammar patterns. This likely also increases the number of false negatives in the random negatives from other constructs. Overall, the classification quality seems near optimal given the quality of the augmented data, which sets an upper performance bound. Due to eliminating duplicates and having 25% random examples from other constructs in the validation set, the accuracy can exceed the 87.1% example accuracy from the manual evaluation.

4.3 Grammar-Controlled Text Generation

Table 3 lists the average ratio of detected grammar constructions on the given level across all sentences, as detected by the trained classifiers from the previous step.

The two LLM baselines, which employ pure prompting, already show improvements over the static baseline of CEFR-annotated texts. GPT4 increases the frequency of EGP entries on all requested levels. The Mistral baseline shows less pronounced improvements and fails to increase the number of grammar constructs on levels A2 and C2. Generally, the pre-trained models have difficulties using more constructs of the levels A2, B1, and B2. Our approach to ranking sentence candidates during text generation has a severe positive impact on the distribution of grammar constructions across all six levels. For all levels, the ratio of applied grammar rules has at least doubled, on level C1 it has even quadrupled versus the baseline. This proves that the variance within different generated candidates regarding the grammatical constructions is sufficient, although the prompt included the instruction to control text complexity. Figure 5 provides a bigger picture of the generated text characteristics between the static baseline versus our method.

While the grammatical constructions in the corpus are much more evenly distributed across all text difficulties, our ranking approach can create visible spikes on the desired complexity level while roughly maintaining the frequencies of other levels' patterns. Only on requested level B2, constructs of level C1 are also increased which may even help scaffolding. Overall, the intervention seems to help control the desired pedagogical properties of generated text.

5 Discussion and Conclusion

This work showcases how Large Language Models can be controlled based on the qualitative EGP augmented to a large-scale dataset to align with pedagogical use cases, specifically – but not limited to – language teaching. We first verified the sufficient quality of LLM-generated instances of an established grammar repository, the English Grammar Profile. The validation emphasizes the strength of the most recent closed-source model, GPT4. Nevertheless, the quality of instances generated by GPT3.5 could almost keep up with the flagship model. Because of this positive finding, we generated 946K labeled grammar construction examples, which we publicly share for further research. The binary grammar construction classification on this data shows satisfying results within the distribution

Table 3: Ratio of detected constructs by CEFR level of the corresponding texts on the same level.

Method	A1↑	A2↑	B1↑	B2↑	C1↑	C2↑
Static Baseline	13.4%	5.0%	5.2%	3.9%	5.1%	5.1%
GPT4	22.2%	5.7%	7.0%	7.3%	14.2%	10.9%
Mistral Prompting	16.1%	4.2%	6.1%	6.5%	9.7%	4.6%
Mistral Candidate Ranking (Ours)	**29.6%**	**10.0%**	**12.3%**	**12.2%**	**20.6%**	**11.3%**

of generated data. The results are close to similar research that has not used minimal pairs and shared embedding models and solved a potentially easier problem (Okano et al., 2023). Controlling an open LLM such as Mistral on the used grammar constructions with these classifiers significantly affects the frequency of desired grammar patterns. It can even beat the baseline of prompting GPT4. While the prompt-based strategies already improve over the static baseline for most CEFR levels, our proposed approach has improved text on every proficiency level and at least doubled the default frequency of constructs on all levels. It also solves the shortcoming of Tyen et al. (2022) that had difficulties generating text on the simpler levels A1 and A2.

With the advent of performant open LLMs, such as Llama3 and Mixtral of Experts, educational applications can be further tailored to align with pedagogical expectations than with prompting alone. Currently, instructors can only use commercial model interfaces such as ChatGPT or third-party wrappers around the model endpoints. Our method advances the possibilities from prompt engineering approaches to fine adjustment of the model output. We freely release our augmented dataset and the trained grammar classifiers to provide learning engineers with a tool to introduce this level of control to their applications. A possible application is adjusting the grammatical complexity of an AI tutor in science to the language proficiency level of each student. Non-native speakers in the same classroom can interact with the seemingly "same" agent that adapts its language to them under the hood. Language instructors can use models to generate texts of students' interests while ensuring the use of particular grammar that aligns with their curriculum.

5.1 Ethical considerations

The EGP was created and annotated by experts to empirically identify the characteristics of the English used by learners at different levels of proficiency. While the data stems from official proficiency tests taken by a wide variety of language learners worldwide, the language used may still be biased by the test tasks, the opinions expressed by the learners who took the tests, and the selection of learner data selected as examples for the EGP. Instructing the LLM to focus on grammatical structure instead of content should mitigate such bias in the generated dataset, though this is not guaranteed. The grammar classification may thus work better for topics typically used by a specific student subgroup in particular language tasks. The authors also acknowledge the potential critique of the CEFR classification as eurocentric (O'Keeffe and Mark, 2017).

Another consideration related to the use of LLMs is the potential generation of toxic or biased language, which is especially sensitive when underage students are working with an LLM-based language learning tool (Meyer et al., 2023). On the pedagogical side, the use of artificially generated text may also limit authenticity and thereby reduce learner motivation. Finally, interacting with a machine to foster language acquisition will not offer the same social benefits and challenges as human interaction.

5.2 Future Work

Future work should build real applications for the educational text generation approach. Then, a controlled field experiment should be pursued to assess the impact on students' language acquisition. It should survey the perception of the generated texts by teachers and students and measure learning gains. This may reveal details about potential weaknesses and issues for example with lexical complexity for which our approach does not explicitly control. With more invasive adaptation techniques, the approach can be easily extended to single grammar constructions and adapt grammar not only to the holistic proficiency level of the learner but to the knowledge of single grammar patterns. The grammar constructions should be further located within the sentences to increase the detection quality and enable annotations. This enables more precise input enhancement applications.

6 Limitations

The training data for grammar classification has some drawbacks. Having only many positive and negative examples is likely insufficient for robust control over single grammar patterns in educational text generation. The models usually use the same sentence structure for creating new instances, especially given the scarcity of seed examples in the EGP. Although the classifiers can learn most of the differences within the generated dataset, it remains unclear how well the classifiers generalize to other models' generative distributions or real-world corpora. More diverse examples must be fostered, and a manual validation of grammar construction detection in real corpora would be needed.

In the text generation step, we only maximized the amount of constructs on the desired CEFR level. A suitable text likely also requires reducing the number of overly difficult constructs to not confuse the learner and better target the zone of proximal development. One could add a parameter that balances how large the penalty for the presence of more difficult grammar should be. Furthermore, some grammar patterns may occur too infrequently in sampling from a pre-trained model, and generating many candidates to obtain at least one positive instance would be inefficient. This can only be overcome by adapting the weights of the pre-trained language model or advanced decoding strategies. Therefore, we tested the approach only on the six groups of grammar construction, as given by their CEFR level, which limits the current approach to less fine-grained control over text generation. However, we believe this can still serve as a proof of concept.

We are also aware that the English Grammar Profile is a description of the typical proficiency level when learners start to use a grammar pattern. This can only serve as a proxy for reading comprehension, which is the focus of this work. Fortunately, our grammar classifiers can serve to analyze existing materials that are expert-curated to create a valid mapping to reading comprehension levels instead of written production.

7 Acknowledgements

We gladly acknowledge the use of the EGP using their officially requested statement: This publication has made use of the English Grammar Profile. This resource is based on extensive research using the Cambridge Learner Corpus and is part of the English Profile program, which aims to provide evidence about language use that helps to produce better language teaching materials. See `http://www.englishprofile.org` for more information.

References

Karin Berendes, Sowmya Vajjala, Detmar Meurers, Doreen Bryant, Wolfgang Wagner, Maria Chinkina, and Ulrich Trautwein. 2018. Reading demands in secondary school: Does the linguistic complexity of textbooks increase with grade level and the academic orientation of the school track? *Journal of Educational Psychology*, 110(4):518–543.

Xiaobin Chen, Detmar Meurers, and Patrick Rebuschat. 2022. ICALL offering individually adaptive input: Effects of complex input on L2 development. *Language Learning and Technology*, 26(1):1–21.

Kiyomi Chujo, Kathryn Oghigian, and Shiro Akasegawa. 2015. A corpus and grammatical browsing system for remedial EFL learners. In Agnieszka Leńko-Szymańska and Alex Boulton, editors, *Studies in Corpus Linguistics*, volume 69, pages 109–128. John Benjamins Publishing Company, Amsterdam.

{Council of Europe}. 2020. *Common European Framework of Reference for Languages: Learning, teaching, assessment – Companion volume*. Council of Europe Publishing, Strasbourg.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Nick C. Ellis. 2002. FREQUENCY EFFECTS IN LANGUAGE PROCESSING: A Review with Implications for Theories of Implicit and Explicit Language Acquisition. *Studies in Second Language Acquisition*, 24(2):143–188.

Alex Housen, Folkert Kuiken, and Ineke Vedder. 2012. *Dimensions of L2 Performance and Proficiency*. John Benjamins Publishing Company, Amsterdam.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger P. Levy. 2020. A Systematic Assessment of Syntactic Generalization in Neural Language Models.

Katharina Jeblick, Balthasar Schachtner, Jakob Dexl, Andreas Mittermeier, Anna Theresa Stüber, Johanna Topalis, Tobias Weber, Philipp Wesp, Bastian Oliver Sabel, Jens Ricke, and Michael Ingrisch. 2023. ChatGPT makes medicine easy to swallow: an exploratory case study on simplified radiology reports. *European Radiology*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B.

Enkelejda Kasneci, Kathrin Sessler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, Stephan Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, Matthias Stadler, Jochen Weller, Jochen Kuhn, and Gjergji Kasneci. 2023. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103:102274.

Osama Koraishi. 2023. Teaching English in the age of AI: Embracing ChatGPT to optimize EFL materials and assessment. *Language Education and Technology*, 3(1).

Stephen Krashen. 1992. The input hypothesis: An update. *Linguistics and language pedagogy: The state of the art*, pages 409–431.

Henry Kučera, Winthrop Francis, William Freeman Twaddell, Mary Lois Marckworth, Laura M. Bell, and John Bissell Carroll. 1967. *Computational analysis of present-day American English*. Brown University Press, Providence.

Karen Lichtman and Bill VanPatten. 2021. Was Krashen right? Forty years later. *Foreign Language Annals*, 54(2):283–305.

Shawn Loewen. 2021. Was Krashen right? An instructed second language acquisition perspective. *Foreign Language Annals*, 54(2):311–317.

Cagri Tugrul Mart. 2013. Teaching grammar in context: why and how? *Theory & Practice in Language Studies*, 3(1):124–129.

Jesse G. Meyer, Ryan J. Urbanowicz, Patrick C. N. Martin, Karen O'Connor, Ruowang Li, Pei-Chen Peng, Tiffani J. Bright, Nicholas Tatonetti, Kyoung Jae Won, Graciela Gonzalez-Hernandez, and Jason H. Moore. 2023. ChatGPT and large language models in academia: opportunities and challenges. *BioData Mining*, 16(1):20.

Yuki Okano, Kotaro Funakoshi, Ryo Nagata, and Manabu Okumura. 2023. Generating Dialog Responses with Specified Grammatical Items for Second Language Learning. In *BEA 2023*, pages 184–194, Toronto, Canada. ACL.

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.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model.

Jiao Sun, Yufei Tian, Wangchunshu Zhou, Nan Xu, Qian Hu, Rahul Gupta, John Wieting, Nanyun Peng, and Xuezhe Ma. 2023. Evaluating Large Language Models on Controlled Generation Tasks. In *EMNLP 2023*, pages 3155–3168, Singapore. ACL.

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.

Leonie Weissweiler, Valentin Hofmann, Abdullatif Köksal, and Hinrich Schütze. 2022. The better your Syntax, the better your Semantics? Probing Pretrained Language Models for the English Comparative Correlative. In *EMNLP 2022*, pages 10859–10882, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jerry Wellington and Jonathan Osborne. 2001. *Language and Literacy in Science Education*. McGraw-Hill Education, UK.

Changrong Xiao, Sean Xin Xu, Kunpeng Zhang, Yufang Wang, and Lei Xia. 2023. Evaluating Reading Comprehension Exercises Generated by LLMs: A Showcase of ChatGPT in Education Applications. In *BEA 2023*, pages 610–625.

Kevin P. Yancey, Geoffrey Laflair, Anthony Verardi, and Jill Burstein. 2023. Rating Short L2 Essays on the CEFR Scale with GPT-4. In *BEA 2023*, pages 576–584, Toronto, Canada. ACL.

Danni Yu, Luyang Li, Hang Su, and Matteo Fuoli. 2023. Assessing the potential of AI-assisted pragmatic annotation: The case of apologies.

Azrifah Zakaria, Willy A. Renandya, and Vahid Aryadoust. 2023. A Corpus Study of Language Simplification and Grammar in Graded Readers. *LEARN Journal: Language Education and Acquisition Research Network*, 16(2):130–153.

LLMs in Short Answer Scoring: Limitations and Promise of Zero-Shot and Few-Shot Approaches

Imran Chamieh[1], Torsten Zesch[2], Klaus Giebermann[1]
[1]Hochschule Ruhr West, Germany,
[2]CATALPA, FernUniversität in Hagen, Germany

Abstract

This study investigates the potential of Large Language Models (LLMs), in particular GPT and LLaMA, for automated scoring of short answer responses. We focus on zero-shot and few-shot settings, but also compare with fine-tuned models and a supervised upper-bound. Our results show that LLMs perform much worse in those settings on a performance level that is not feasible for practical purposes. Fine-tuning LLMs brings their results on roughly the same level as supervised results, but as they are less efficient there currently seems to be no basis for applying LLMs for short answer scoring.

1 Introduction

The constantly increasing demand placed on educators in today's educational landscape requires innovative solutions to replace traditional assessment methods. Manual assessment, especially for large-scale exams, presents challenges for scalability, consistency and timely feedback to students Ramesh and Sanampudi (2022). Automated scoring has emerged as a potential solution, promising faster, more objective and feedback-rich assessments Galhardi and Brancher (2018).

Extensive research has explored automated scoring, but many systems require large amounts of training data to achieve reliable performance Patil and Adhiya (2022). Our focus is on finding a system that demonstrates strong performance across different datasets while minimizing the need for huge number of training examples. Large Language Models (LLMs) seem promising in this regard Naveed et al. (2023). Thus, in this paper, we explore LLMs performance in scoring open-ended student answers across three datasets. We compare two prominent LLMs, Generative Pre-trained Transformer (GPT) and Large Language Model Meta AI (LLaMA), under different training settings, including zero- and few-shot learning, and fine-tuning specifically applied to the GPT model. Additionally, we benchmark their performance against established baselines, specifically Google's pre-trained language model BERT Devlin et al. (2018) and classical SVM , known for its robustness in classification tasks Cortes and Vapnik (1995). This evaluation aims to deepen our understanding of how LLMs handle various assessment tasks and shed light on their potential to enhance automated scoring in education, particularly with limited training data.

2 Related Work

Very few studies have explored the performance of LLMs in zero- and few-shot settings within the context of automated scoring. Wu et al. (2023) introduced the Matching exemplars as Next Sentence Prediction (MeNSP) method, by employing a zero-shot prompt learning method using pre-trained language models. Their results indicate that few-shot learning offered limited improvement in performance.

Latif and Zhai (2024) compare the performance of a fine-tuned GPT-3.5 model with BERT and demonstrated that GPT-3.5 achieved higher scoring accuracy. It showed a remarkable average increase of 9.1% compared to BERT when applied to a single dataset of six assessment tasks. This finding emphasized the need for domain-specific fine-tuning LLMs to enhance their performance.

On the other hand, many studies investigated the neural networks and machine learning models to build scoring tools. Steimel and Riordan (2020) demonstrate how pretrained transformer models could be adapted for content scoring using an instance-based approach. By pooling token representations across all model layers, this approach achieved state-of-the-art performance on short answer scoring tasks. Bexte et al. (2023) conduct a comparison between instance-based and similarity-based methods on multiple datasets. They investigated the influence of different training set sizes

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 309–315
June 20, 2024 ©2024 Association for Computational Linguistics

on the performance of these methods using learning curve experiments. It found that a fine-tuned SBERT model does often yield the best results.

Overall, existing research offers limited insight into how LLMs perform in zero-shot and few-shot settings.

3 Experimental Setup

We tested the GPT family of models introduced by OpenAI, specifically GPT-3.5 and GPT-4.[1] Additionally, we tested Meta AI's LLaMA-2 models LLaMA-7b, LLaMA-13b, and LLaMA-70b [2]. Finally, Google's BERT model and SVM were included as baselines for comparison. For testing, we randomly selected 20% of each task from the datasets. We observed that LLMs usually produce in addition to the score an explanation, or repeat the scores of the given shots, rather than providing only the score, so we applied a filtering function that retrieves only the last integer of the LLMs response, and if no integer was found, we assigned a randomly generated number between 0 and the maximum possible score of the current task.

3.1 Datasets & Evaluation

We performed experiments on three widely used answer scoring datasets that are freely available.

ASAP Automated Student Assessment Prize[3] contains 10 prompts covering a broad range of disciplines. All answers were scored by two humans on a 0-2 or 0-3 scale depending on the task.

MindReading contains responses from children (ages 7-13) on questions from the Strange Story and Silent Film tasks, where answers scored on a 0-2 scale Kovatchev et al. (2020).

Powergrading is a short-answer dataset focused on knowledge about the United States for the citizenship exam. Answers are scored on a 0-1 scale Basu et al. (2013).

In this study, we differentiate between the terms 'Task' and 'Prompt'. 'Task' refers to a specific question from the datasets used. While,'Prompt' is a set of instructions designed for the LLM, including scoring guidelines, relevant context, and the student answer to be scored. For few-shot model, the prompt also includes randomly selected answer samples for each score within the task of the studied dataset.

For each task, we calculated Quadratic Weighted Kappa (QWK) Cohen (1968) as a standard metric used to quantify the agreement between machine scoring and human expert scoring. Finally, we averaged QWK scores across all tasks, for each dataset, to obtain a single overall performance metric.

3.2 Prompting

For the Powergrading dataset and 5 tasks within the ASAP dataset, we explored zero-shot performance. Note that zero-shot was not suitable for other ASAP tasks due to their reliance on long-form text or image data, nor for the MindReading dataset, where the questions are unavailable. To investigate the effectiveness of few-shot model for score prediction, we employed a variety of prompt designs and evaluated them on different numbers of shots. Initial testing (1, 3, 5, and 10 shots) with three prompt designs – Newline, Semicolon, and Space delimiters – revealed minimal variance in results, unlike to what Sclar et al. (2023) found (see Appendix B). Based on these results, we proceeded with the Newline delimiter prompt design for subsequent experiments from 0 to 10 shots, as it showed consistent performance across initial tests.

3.3 Fine-tuning

We extended our study to unveil the potential of the LLMs by fine-tuning a GPT-3.5 model. Fine-tuning involves adjusting the pre-trained model's parameters to adapt to specific characteristics of the task under study. For training phase we used 80% of the data to fine-tune GPT-3.5-turbo-1106.

4 Results & Discussion

Table 1 gives an overview of our results. The supervised system is a reference point that we use to compare zero-shot and few-shot result with.

4.1 Fine-tuning

Contrary to the results in Latif and Zhai (2024), which was conducted on a limited dataset, our results in Table 1 show that BERT actually scores slightly higher QWK over all datasets. This suggests a potential overfitting issue in GPT-3.5 model. In particular, in three tasks of Powergrading dataset, we observed that the model consistently scored all answers as 1. In general, fine-tuned results are in the same ballpark as supervised results, but computationally much more expensive.

[1] https://openai.com/

[2] https://ai.meta.com/

[3] https://www.kaggle.com/c/asap-sas

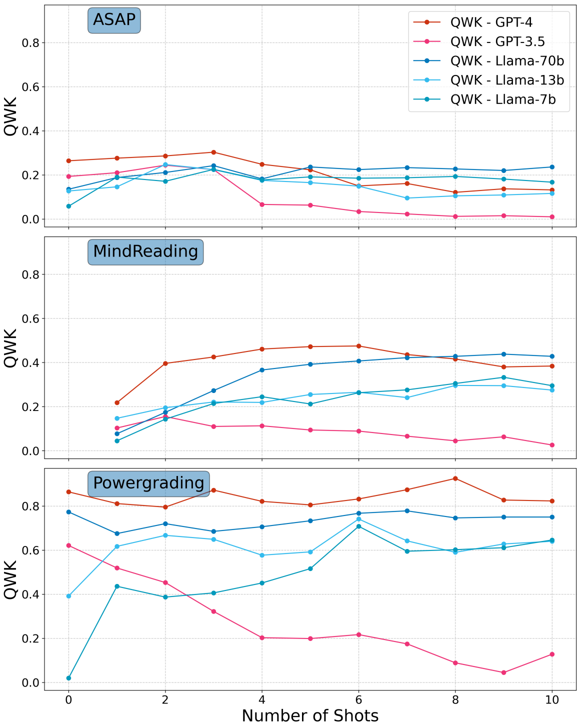


Figure 1: Impact of number of shots on scoring performance

4.2 Zero-shot

LLMs performance in zero-shot settings varied significantly across datasets. GPT-4 showed promising results on Powergrading dataset, while all models performed poorly on ASAP dataset. This suggests that LLMs are not yet mature enough for reliable zero-shot automated scoring. The good performance on Powergrading dataset can be attributed to the simplicity of the questions, which are related to USA citizenship test, and the scoring range (0,1). In contrast, even with detailed prompt and rubrics scoring instructions, LLMs struggled with ASAP dataset, indicating their limitations on tasks that require complicated reasoning or relay on domain-specific knowledge.

4.3 Few-shot

Our initial expectation was that incorporating few-shot into the prompt would enhance the model performance, as observed in the previous study Wu et al. (2023). However, our results indicate that only LLaMA models on Powergrading and MindReading datasets showed a slight improvement in performance with an increasing number of shots (up to 6 shots). In contrast, GPT-3.5 exhibited a weird behavior, with performance decreasing as the number of shots increasing, in particular on Powergrading dataset.

		QWK		
		ASAP	MR	PG
supervised	BERT	.74	.87	.94
	SVM	.46	.74	.80
fine-tuning	GPT-3.5	.61	.81	.83
0-shot	GPT-4	.26	.22	.86
	GPT-3.5	.19	.10	.62
	LLaMA-70b	.14	.08	.77
	LLaMA-13b	.13	.15	.39
	LLaMA-7b	.06	.05	.02
3-shot	GPT-4	.30	.43	.87
	GPT-3.5	.22	.11	.32
	LLaMA-70b	.24	.27	.69
	LLaMA-13b	.22	.22	.65
	LLaMA-7b	.23	.21	.41

Table 1: Comparison of model performance in terms of Quadratic Weighted Kappa (QWK)

The poor performance of LLMs in ASAP dataset is attributed to two key factors. First, answers in ASAP dataset tend to be longer compared to answers in other datasets, as shown in Figure 2 (see Appendix), where the average length of answers is approximately 50 words, so adding few-shot for each score increases the prompt size rapidly, which might badly affect the output. Additionally, the dataset's complexity, as questions heavily depend on domain-specific knowledge indicates challenges for general models in such domains. Similarly, in MindReading dataset, not only the questions are not available, but these questions are also derived from strange stories or silent films and they rely on specific knowledge that LLMs may not be trained on. On the other hand, the questions presented on Powergrading dataset related to general knowledge about USA, which made it easy for the LLMs to predict the scores which were limited to 0 and 1. Additionally, the short length of answers enables LLMs to effectively memorize it's task, making score prediction easier.

5 Conclusion

This study explores the potential of LLMs in automated scoring tasks, specifically zero- and few-shot, and fine-tuned settings across three diverse datasets.

Overall, our findings reveal strong performance from zero-shot and few-shot models on general knowledge. GPT-4 achieved performance very close to the upper bound BERT and outperformed SVM model. LLaMA models showed promising results; while not reaching GPT-4 levels, their performance remained consistent across different numbers of shots. In contrast, GPT-3.5 appeared overfitting as more shots introduced. This highlights the potential of few-shot LLMs for short answer scoring, especially on tasks involving general knowledge questions.

However, LLMs face challenges when confronted with tasks that require complicated reasoning or domain-specific knowledge, as noticed by their poor performance in ASAP dataset. The complicated nature of the questions in these subjects appears to cause difficulties for LLMs, highlighting the need for further improvements in dealing with nuanced and specialized content within educational datasets.

With regard to the fine-tuned model, our study revealed unexpected results as it failed to meet our performance expectations for automated scoring. It became clear that the model was overfitting at certain questions 'tasks', particularly noticeable when examining the performance of the Powergrading dataset.

Limitations

We only test commercial LLMs, but argue that open source LLMs would very likely yield even worse results. So the overall conclusion of the paper that LLMs are not yet ready to be used in zero-shot or few-shot settings for short answer scoring would stand unchanged. However, in future work we want to test a wider range of LLMs to gain further insights into their capabilities.

Ethical Considerations

LLMs are trained on large data sets that may contain unintentional biases, potentially leading to unfair scoring based. LLMs as black-box can lack transparency, making it difficult to provide an interpretation how they predict the scores. Another significant concern is student data privacy. If an LLM is hosted online, student answers are send to the provider and could end up in the model training data.

References

Sumit Basu, Chuck Jacobs, and Lucy Vanderwende. 2013. Powergrading: a clustering approach to amplify human effort for short answer grading. *Transactions of the Association for Computational Linguistics*, 1:391–402.

Marie Bexte, Andrea Horbach, and Torsten Zesch. 2023. Similarity-based content scoring-a more classroom-suitable alternative to instance-based scoring? In *Findings of the association for computational linguistics: Acl 2023*, pages 1892–1903.

Jacob Cohen. 1968. Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

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*.

Lucas Busatta Galhardi and Jacques Duílio Brancher. 2018. Machine learning approach for automatic short answer grading: A systematic review. In *Advances in Artificial Intelligence-IBERAMIA 2018: 16th Ibero-American Conference on AI, Trujillo, Peru, November 13-16, 2018, Proceedings 16*, pages 380–391. Springer.

Venelin Kovatchev, Phillip Smith, Mark Lee, Imogen Grumley Traynor, Irene Luque Aguilera, and Rory T Devine. 2020. " what is on your mind?" automated scoring of mindreading in childhood and early adolescence. *arXiv preprint arXiv:2011.08035*.

Ehsan Latif and Xiaoming Zhai. 2024. Fine-tuning chatgpt for automatic scoring. *Computers and Education: Artificial Intelligence*, page 100210.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Nick Barnes, and Ajmal Mian. 2023. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.

Shweta Patil and Krishnakant P Adhiya. 2022. Automated evaluation of short answers: A systematic review. *Intelligent Data Communication Technologies and Internet of Things: Proceedings of ICICI 2021*, pages 953–963.

Dadi Ramesh and Suresh Kumar Sanampudi. 2022. An automated essay scoring systems: a systematic literature review. *Artificial Intelligence Review*, 55(3):2495–2527.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2023. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *arXiv preprint arXiv:2310.11324*.

Kenneth Steimel and Brian Riordan. 2020. Towards instance-based content scoring with pre-trained transformer models. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, volume 34.

Xuansheng Wu, Xinyu He, Tianming Liu, Ninghao Liu, and Xiaoming Zhai. 2023. Matching exemplar as next sentence prediction (mensp): Zero-shot prompt learning for automatic scoring in science education. In *International Conference on Artificial Intelligence in Education*, pages 401–413. Springer.

A Models Hyperparameters

1. **SVM with TF-IDF vectorization:** We used a linear SVM model with a fixed Regularization parameter C=1.0 and utilized TF-IDF vectorization with a maximum vocabulary size of 1000 features.

2. **BERT:** We used the pre-trained BERT ("bert-base-uncased") model. Training data is processed with the BertTokenizerFast tokenizer and padded to a uniform length (512). we trained the model for 20 epochs with batches size = 8. After each epoch we run evaluation and kept the model with the lowest validation loss for evaluation on testing data.

3. **GPT:** For fine-tuning we utilized the OpenAI-recommended GPT-3.5-turbo-1106 model. Where as, GPT-4 is not yet available for fine-tuning. Training, validation, and test data were formatted in JSONL files as required. We employed the default values (auto) for learning rate, num_epochs, and batch_size.
   For few-shot experiments both GPT-3.5-turbo and GPT-4-turbo-preview were tested using the default parameters.

4. **LLaMA:** We utilized LLaMA-7b, LLaMA-13b, and LLaMA-70b for the few-shot model with the following parameter:
   temperature: 0.6 (Adjusts randomness of outputs. Higher values increase randomness, lower values promote determinism.). **top_p:** 0.9 (Controls text generation. Samples from the top 90% of most likely tokens, allowing for some variation.). **max_seq_len:** we choose different values between 512 and 2056, depending on the dataset and number of shots (It refers to the maximum length of input sequences the model can process.). **max_gen_len:** 5 as we want only the score. (It sets a limit on the maximum length of generated responses.). **max_batch_size:** int = 4

B Prompt designs

New line delimiter Evaluate student response to the United States Citizenship Exam. Return only the score, 1 if it is correct, and 0 if it is wrong. Question: What is one right or freedom from the First Amendment? (Return only the score):

Answer: the right to assemble -> Score: 1

Answer: freedom of speech -> Score: 1

Answer: freedom of religion.s -> Score: 1

Answer: right to pursue happiness. -> Score: 0

Answer: right to bear arms -> Score: 0

Answer: privacy -> Score: 0

Answer: free speech -> Score:

Semicolon delimiter Evaluate student response to the United States Citizenship Exam. Return only the score, 1 if it is correct, and 0 if it is wrong. Question: What is one right or freedom from the First Amendment? (Return only the score): Answer: the right to assemble -> Score: 1; Answer: freedom of speech -> Score: 1; Answer: freedom of religion.s -> Score: 1; Answer: right to pursue happiness. -> Score: 0; Answer: right to bear arms -> Score: 0; Answer: privacy -> Score: 0; Answer: free speech -> Score:

Space delimiter Evaluate student response to the United States Citizenship Exam. Return only the score, 1 if it is correct, and 0 if it is wrong. Question: What is one right or freedom from the First Amendment? (Return only the score): Answer: the right to assemble -> Score: 1 Answer: freedom of speech -> Score: 1 Answer: freedom of religion.s -> Score: 1 Answer: right to pursue happiness. -> Score: 0 Answer: right to bear arms -> Score: 0 Answer: privacy -> Score: 0 Answer: free speech -> Score:

	1 Shot		3 Shot		5 Shots		10 Shots	
Dataset	**qwk**	**Acc**	**qwk**	**Acc**	**qwk**	**Acc**	**qwk**	**Acc**
ASAP	**.152**	.417	.068	.319	.057	.287	.029	.234
	.077	.349	.051	.309	.034	.256	.010	.198
	.105	.355	**.083**	.343	**.058**	.288	**.037**	.223
Mindreading	.217	.516	.137	.419	**.133**	.400	.101	.403
	.237	.534	.128	.436	.122	.423	.104	.443
	.195	.500	**.139**	.423	**.133**	.420	**.108**	.428
Powergrading	**.656**	.909	**.077**	.500	.105	.767	**.156**	.793
	.403	.832	.058	.538	**.320**	.876	.098	.842
	.373	.843	.028	.507	.199	.838	.149	.813

Table 2: Impact of Delimiters (New Line, Semicolon, Space) of prompt on Accuracy and QWK using GPT-3.5

	1 Shot		3 Shots		5 Shots	
Dataset	**qwk**	**Acc**	**qwk**	**Acc**	**qwk**	**Acc**
ASAP	.244	.465	**.280**	.486	.220	.453
	.250	.460	.264	.482	.230	.487
	.244	.452	.250	.472	**.233**	.478
Mindreading	.133	.415	.408	.625	.448	.643
	.136	.416	**.447**	.655	**.507**	**.688**
	.129	.411	.409	.631	.476	.673
Powergrading	**.788**	.941	**.798**	.947	.777	.934
	.774	.940	.794	.941	.749	.922
	.781	.945	.794	.946	**.786**	.927

Table 3: Impact of Delimiters (New Line, Semicolon, Space) of prompt on Accuracy and QWK using GPT-4

	1 Shots		3 Shots		5 Shots		10 Shots	
Dataset	**qwk**	**Acc**	**qwk**	**Acc**	**qwk**	**Acc**	**qwk**	**Acc**
ASAP	**.190**	.465	**.225**	.503	**.190**	.483	**.168**	.455
	.154	.448	.195	.463	.137	.414	.112	.400
	.161	.462	.219	.477	.157	.448	.137	.429
Mindreading	.043	.190	.206	.523	.213	.517	**.296**	.565
	.094	.455	**.252**	.578	.193	.541	.212	.487
	.082	.436	.236	.569	**.274**	.588	.289	.540
Powergrading	.324	.698	.390	.709	**.500**	.829	**.646**	.901
	.278	.751	**.460**	.808	.461	.817	.583	.872
	.334	.732	.452	.806	**.500**	.801	.572	.866

Table 4: Impact of Delimiters (New Line, Semicolon, Space) of prompt on Accuracy and QWK using LLaMA2-7b-chat

	1 Shots		3 Shots		5 Shots		10 Shots	
Dataset	**qwk**	**Acc**	**qwk**	**Acc**	**qwk**	**Acc**	**qwk**	**Acc**
ASAP	.148	.428	**.237**	.518	.180	.470	**.132**	.430
	.200	.477	.186	.463	.123	.420	.090	.371
	.191	.470	.166	.461	**.184**	.469	.130	.416
Mindreading	.097	.367	**.221**	.505	**.255**	.528	**.285**	.562
	.110	.418	.217	.512	.219	.520	.219	.502
	.097	.395	.198	.476	.215	.491	.282	.557
Powergrading	.541	.848	.547	.818	.573	.862	.656	.884
	.620	.875	**.582**	.828	**.579**	.842	**.712**	.827
	.558	.835	.518	.826	.617	.872	.631	.875

Table 5: Impact of Delimiters (New Line, Semicolon, Space) of prompt on Accuracy and QWK using LLaMA2-13b

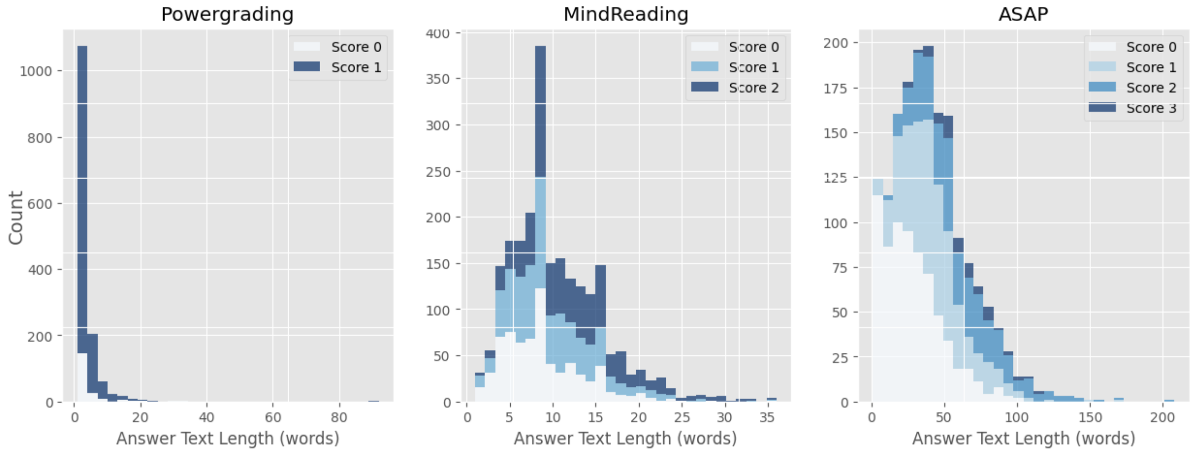


Figure 2: Length distribution of answers

Automated Essay Scoring Using Grammatical Variety and Errors with Multi-Task Learning and Item Response Theory

Kosuke Doi **Katsuhito Sudoh** **Satoshi Nakamura**
Nara Institute of Science and Technology
{doi.kosuke.de8, sudoh, s-nakamura}@is.naist.jp

Abstract

This study examines the effect of grammatical features in automatic essay scoring (AES). We use two kinds of grammatical features as input to an AES model: (1) grammatical items that writers used correctly in essays, and (2) the number of grammatical errors. Experimental results show that grammatical features improve the performance of AES models that predict the holistic scores of essays. Multi-task learning with the holistic and grammar scores, alongside using grammatical features, resulted in a larger improvement in model performance. We also show that a model using grammar abilities estimated using Item Response Theory (IRT) as the labels for the auxiliary task achieved comparable performance to when we used grammar scores assigned by human raters. In addition, we weight the grammatical features using IRT to consider the difficulty of grammatical items and writers' grammar abilities. We found that weighting grammatical features with the difficulty led to further improvement in performance.[1]

1 Introduction

Automated Essay Scoring (AES) is a task that automatically grades essays. Essay assignments are widely used in language tests and classrooms to assess learners' writing abilities, while grading them takes time and effort for human raters. Maintaining inter- and intra-rater reliability is another issue associated with human scoring. AES can help alleviate these problems and has been attracting more attention in recent years.

The grading methods for essays can be roughly categorized into two types: holistic scoring and analytic scoring. The former assigns a single score to an essay based on its overall performance, while the latter assigns different scores to various aspects of the essay, such as grammar, vocabulary, content, or organization (Weigle, 2002). However, rubrics for holistic scoring typically contain descriptions of several aspects of writing used in analytic scoring (*e.g.*, TOEFL iBT Independent Writing Rubric).

Among those aspects, we focus on grammatical features, inspired by the research on criterial features for the levels of the Common European Frameworks of References (CEFR) (Council of Europe, 2001) in L2 English (Hawkins and Filipović, 2012). The CEFR, one of the influential frameworks in language teaching, describes language abilities in functional terms (*i.e.*, can-do statements, such as "Can write short, simple essays on topics of interest"). However, it is grammatical items and lexis that realize the functions written in can-do statements. To fully develop and elaborate their ideas in essays, they need to use a wide range of grammatical items. In fact, grammar plays an important role in essay scoring. Researches on writing in the second language acquisition field have been focusing on syntactic complexity[2] and accuracy (see Kuiken, 2023; Housen et al., 2012).

Hawkins and Filipović (2012) identified grammatical items that learners at a certain level and higher can use correctly and items that learners at a certain level are prone to making mistakes in. It is known that human raters look for those features consciously or unconsciously when they evaluate learners' performance, and explicit use of grammatical features in AES will improve model performance.

Grammatical features have been used in many feature-engineering AES models (see Ke and Ng, 2019) as well as in hybrid models, which incorporate handcrafted features into deep neural network AES models (Dasgupta et al., 2018; Uto et al., 2020; Bannò and Matassoni, 2022). In

[1] The code is publicly available at https://github.com/ahclab/aes-grammar-mtl-irt.

[2] Syntactic complexity refers to the extent to which a learner can use a wide variety of both basic and sophisticated structures (Wolfe-Quintero et al., 1998).

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 316–329
June 20, 2024 ©2024 Association for Computational Linguistics

Yannakoudakis et al. (2011), features representing grammatical structures were used together with other linguistic features. However, in many previous studies (*e.g.*, Vajjala, 2018; Uto et al., 2020), grammatical items used correctly were aggregated into measures of grammatical complexity (*e.g.*, ratio of dependent clauses per clauses; see Wolfe-Quintero et al., 1998) rather than individual grammatical items (*e.g.*, adverbial clause *if*, adverbial clause *so that*) even though the difficulties of individual grammatical items are different.

In this paper, we propose to use individual grammatical items as inputs to hybrid AES models that predict holistic scores, and leverage the models to incorporate the variety of grammatical items in grading essays. We also use frequencies of grammatical errors corrected by a modern grammatical error correction model (GECToR-large; Tarnavskyi et al., 2022) as model inputs. The grammatical features are combined with an essay representation and passed into a fully connected feed-forward neural network to predict the score of an input essay. Our models used BERT (Devlin et al., 2019) to learn essay representations following the current state-of-the-art AES models (Yang et al., 2020; Cao et al., 2020; Wang et al., 2022).

To utilize grammatical features more effectively, we develop a multi-task learning framework that jointly learns to predict holistic scores and grammar scores of essays. We use two types of grammar scores: (1) scores assigned to essays by human raters and (2) writers' latent abilities estimated based on patterns of grammar usage using Item Response Theory (IRT) (Lord, 1980). Note that teacher labels are not necessary for estimating the latent abilities using IRT.

IRT estimates not only each writer's ability but also the characteristics of each item (*i.e.*, individual grammatical item), such as discrimination and difficulty parameters. Therefore, we use these IRT parameters to weight grammatical items (*e.g.*, award writers who use a difficult grammatical item; see Section 3.1.2).

In summary, the contributions of this paper are as follows:

- We propose to use individual grammatical items and grammatical errors as inputs to AES models, and leverage the models to consider grammar use in predicting holistic scores of essays.
- We develop a multi-task learning framework that jointly learns to predict holistic scores and grammar scores of essays.
- We apply IRT to writers' grammar usage patterns and (1) use estimated latent abilities for multi-task learning, and (2) use IRT parameters to weight grammatical items when we feed them to AES models.
- We show the effectiveness of incorporating grammatical features into BERT-hybrid AES models. Our method shows a significant advantage on some essay assignments in the Automated Student Assessment Prize (ASAP) dataset[3].

2 Related Work

2.1 Automated Essay Scoring

Early AES models predict essay scores using hand-crafted features (see Ke and Ng, 2019). For example, e-rater (Burstein et al., 2004) uses 12 features, including grammatical errors and lexical complexity measures. Yannakoudakis et al. (2011) automatically extracted various linguistic features, including grammatical structures, using a parser. These features were weighted and used to train SVM ranking models. Vajjala (2018) reported that measures of grammatical complexity and errors were assigned large weights among various linguistic features.

Recently, a deep neural network-based approach has become popular. AES models based on RNN (Taghipour and Ng, 2016), Bi-LSTM (Alikaniotis et al., 2016), and pre-trained language models (Nadeem et al., 2019; Yang et al., 2020; Cao et al., 2020; Wang et al., 2022) have been proposed. In addition, a hybrid model, which incorporates hand-crafted features into a deep neural network-based model, has been proposed (Dasgupta et al., 2018; Uto et al., 2020; Bannò and Matassoni, 2022).

AES using a large language model has also been explored. Mizumoto and Eguchi (2023) demonstrated that using linguistic features in GPT-3 improved AES performance. Yancey et al. (2023) reported that providing a small number of scoring examples to GPT-4 led to comparable performance to models trained on hundreds of thousands of data based on 85 language features.

This study examines the effect of explicitly considering grammatical features in a hybrid AES

[3] https://www.kaggle.com/c/asap-aes

model by incorporating individual grammatical items as model inputs and weighting them using IRT parameters.

2.2 Multi-Task Learning

Multi-task learning (MTL) (Caruana, 1997) is a method that improves the generalization performance of the main task by training a single model to perform multiple tasks simultaneously. MTL has been used in previous studies in AES, and shown to be effective. Cummins et al. (2016) used MTL to overcome the lack of task-specific data in the ASAP dataset by treating each essay prompt as a different task. Xue et al. (2021) also trained a model jointly on eight different prompts in the ASAP dataset using BERT.

There are also studies that have performed MTL with other NLP tasks. Cummins and Rei (2018) trained an LSTM jointly on grammatical error detection and AES. While the error detection task in Cummins and Rei (2018) required the model to predict whether a particular token was errorful, ones in Elks (2021) require to (1) predict a sentence contains errors or (2) classify tokens by a type of error (*e.g.*, correct, lexical, form). Other auxiliary tasks used in previous studies include morpho-syntactic labeling, language modeling, and native language identification (Craighead et al., 2020), sentiment analysis (Muangkammuen and Fukumoto, 2020), predicting the level of each token (Elks, 2021), and predicting span, type, and quality of argumentative elements (Ding et al., 2023).

In this paper, we train models jointly on holistic scores and grammar scores. This is similar to AES models that predict multiple essay traits simultaneously (Mathias and Bhattacharyya, 2020; Hussein et al., 2020; Mim et al., 2019; Ridley et al., 2021), but the difference between them and ours is that we explicitly incorporate grammatical features to a model, which are related to the score to be predicted.

2.3 Item Response Theory

IRT is a probabilistic model that has been widely used in psychological and educational measurement (Hambleton et al., 1991). An IRT model expresses the probability of a correct response to a test item as a function of the item parameters, which represent the characteristics of the item, and the ability parameter, which represents the ability of the examinee.

Previous studies in AES used IRT to mitigate raters' bias (Uto and Okano, 2021), integrate prediction scores from various AES models (Aomi et al., 2021; Uto et al., 2023), and predict multiple essay traits (Uto, 2021; Shibata and Uto, 2022). These studies employed a multidimensional IRT model since unidimensionality cannot be assumed for the subject to which IRT is applied.

In contrast, we regard individual grammatical items as test items, assuming that whether grammar items are used correctly constitutes grammar ability (*i.e.*, satisfy the assumption of unidimensionality). We model writers' grammar ability using two-parameter logistic model (Lord, 1952), formulated by the following equation:

$$P_{ij}(\theta_i) = \frac{1}{1 + \exp(-Da_j(\theta_i - b_j))} \tag{1}$$

where $P_{ij}(\theta_i)$ is the probability that the writer i with ability θ_i uses the grammatical item j correctly, a_j is the discrimination parameter for item j, and b_j is the difficulty parameter for item j. D is a scaling factor and set to 1.0 in this paper.

3 Proposed Method

3.1 Grammatical Features

The Common European Frameworks of References (CEFR) (Council of Europe, 2001) is an internationally recognized framework for language proficiency. It divides proficiency into six levels ranging from A1 (beginner) to C2 (advanced). Due to the language-neutral nature of the CEFR, what grammatical and lexical properties learners develop across the CEFR levels has been studied language by language.

Such properties (criterial features) in English have been identified by English Profile Programme (Hawkins and Filipović, 2012). Criterial features refer to linguistic properties that are characteristic and indicative of L2 proficiency levels and that distinguish higher levels from lower (*ibid*). They identified positive linguistic features (PFs; grammatical items that learners can use correctly at a certain level and higher) and negative linguistic features (NFs; grammatical items that learners at a certain level are prone to making mistakes in) in relation to the CEFR levels.

Based on the analyses of human raters' grading performance in actual exams, Hawkins and Buttery (2009) have argued that they develop clear intuitions about these properties. We expect that allowing a model to learn grammar representations

Features	Descriptions
type256	256 grammatical items, whether a writer use the items
err54	54 types of errors, # of errors
multiply_b	Modify `type256` with item difficulty
prob	Replace elements in `type256` with the probabilities of using the items correctly
multiply_prob	Weight `type256` with the probabilities
add_prob	Consider both the actual use (`type256`) and the probabilities

Table 1: Grammatical features used in our experiments. The number of errors is relative freq. per 100 words.

using grammatical features would improve the AES performance. Table 1 shows PFs and NFs used in our experiments. The following sections describe them in detail.

3.1.1 Positive Linguistic Features

PFs were extracted using a toolkit for frequency analysis of grammatical items, which is provided by the CEFR-J Grammar Profile (Ishii and Tono, 2018). It extracts 501 grammatical items in text based on regular expressions and calculates the frequencies of them. We converted the frequencies into the 256-dimensional vector (`type256`) based on CEFR-J Grammar Profile for Teachers[4] as $\boldsymbol{g_i} = \{g_{i1}, g_{i2}, ..., g_{i256}\}$. Each dimension corresponds to a grammatical item, and $g_{ij} = 1$ if the writer i used the item j in the essay, and 0 if not.

3.1.2 PFs Weighted using IRT Parameters

Researches on criterial features have shown that learners master more and more grammatical items across the CEFR levels, but `type256` does not consider the difficulties of the items. Therefore, we weight them using the IRT parameters.

We transform g_{ij} in the following four ways:

multiply_b: $g'_{ij} = g_{ij} \times b_j$

prob: $g'_{ij} = P_{ij}(\hat{\theta}_i)$

multiply_prob: $g'_{ij} = g_{ij} \times P_{ij}(\hat{\theta}_i)$

add_prob: $g'_{ij} = \alpha g_{ij} + (1 - \alpha) P_{ij}(\hat{\theta}_i)$

where $\hat{\theta}_i$ is the grammatical ability of the writer i estimated based on the patterns of grammar usage using IRT, and α is a weighting parameter. α was set to 0.5 in this paper.

`multply_b` aims to consider the difficulty of items by multiplying the difficulty parameter for the item. However, writers might not have used some grammatical items because of the essay topic although they had enough abilities to do. Therefore, we use $P_{ij}(\hat{\theta}_i)$, which shows the probability that the writer i with ability $\hat{\theta}_i$ can use the item j correctly. In `prob` g_{ij} is replaced with $P_{ij}(\hat{\theta}_i)$, while in `multiply_prob` and `add_prob` the two values are combined to consider both the ability of writers and the actual use in essays.

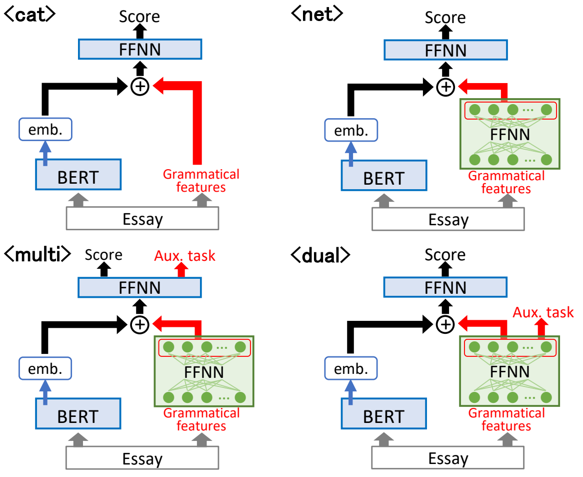


Figure 1: The architectures of proposed models

IRT parameters were estimated independently from the prediction of essay scores. The IRT parameters were frozen during the training of scoring models.

3.1.3 Negative Linguistic Features

We calculated the number of grammatical errors per 100 words as NFs. Specifically, we created the 54-dimensional vector (`err54`) based on error tags assigned by ERRANT (Bryant et al., 2017)[5]. We used GECToR-large (Tarnavskyi et al., 2022) to correct grammatical errors in essays.

3.2 Model Architecture

Our model takes a batch of essays and grammatical features as input and predicts the holistic scores of the essays. We prepare a model that takes only a batch of essays as input for a baseline. Essay representations are obtained from the `[CLS]` token of the BERT model.

Grammatical features are used in the four settings shown in Figure 1. In `cat`, we concatenate the essay representation and the vector of gram-

[4] https://www.cefr-j.org/download.html#cefrj_grammar The toolkit distinguishes the same items in different sentence types such as the affirmative or negative, while CEFR-J Grammar Profile for Teachers does not.

[5] Based on all possible combinations of the error types and categories. We tried the 24-dimensional vector, which was based on the error types, but the 54-dimensional vector improved the model performance more.

matical features, and feed it to a fully connected feed-forward neural network (FFNN). In `net`, we first feed the vector of grammatical features to an FFNN and concatenate the representation from the final layer with the essay representation. In `multi`, we perform multi-task learning with the model architecture of `net`. The FFNN in `multi` consists of shared layers only, and does not have task-specific layers[6]. In `dual`, the predicted values for the auxiliary task are output from the FFNN for grammatical features.

As the labels for the auxiliary task in `multi` and `dual`, we used grammar scores assigned to essays by human raters, which is available in ASAP and ASAP++ dataset (Mathias and Bhattacharyya, 2018), and grammar abilities estimated using IRT. Grammar abilities can be estimated from writers' grammar usage patterns without any teacher labels.

4 Experiments

4.1 Data and Evaluation

We used the ASAP and the ASAP++ dataset in our experiments. The ASAP consists of essays for eight different prompts, with holistic scores for Prompts 1-6 and analytic scores for Prompts 7-8. In Prompt 7 and 8, the weighted sum of the analytic scores constitutes the total score, which is the target of prediction by our models. ASAP++ includes analytic scores of essays for Prompt 1-6. We developed AES models that predict the holistic score for each essay prompt. From analytic scores, we only used ones related to grammar[7].

We evaluated the scoring performance of our models using the Quadratic Weighted Kappa (QWK) on the ASAP dataset. Following the previous studies, we adopted 5-fold cross validation with 60/20/20 split for train, development, and test sets, which was provided by Taghipour and Ng (2016).

4.2 Settings

As explained in Section 3.2, we developed our AES models based on BERT. We used `bert-base-uncased` provided by Hugging Face[8]. The maximum input length was set to 512.

We normalized essay scores in the range of $[-1, 1]$. The mean squared error (MSE) loss was employed for both the main and auxiliary tasks. We updated the parameters for the FFNN and the BERT layers. The number of hidden layers in the FFNN for grammatical features was set to 3, and the number of the nodes in the hidden layer to one-half the dimension of the grammatical features. The number of hidden layers in the FFNN on the top was set to $\{1, 2, 3, 4, 5, 7, 10\}$ for `cat`, $\{1, 2, 3\}$ for `net`, and $\{2, 3\}$ for `multi` and `dual` and we chose the value that achieved the best QWK score on the development set for Prompt 1. The number of the nodes was set to 512. For both FFNNs, we adopted relu as the activation function and set the dropout ratio to 0.2. In `multi` and `dual`, we tried $\{0.8, 0.6\}$ for the weights of the loss function for the main task. We used Adam optimizer (Kingma and Ba, 2015) with a learning rate of 1e-5. We trained models with the batch size $\{4, 8, 16, 32\}$ for 10 epochs. In the following sections, we report the scores on test sets for the batch size with the highest QWK on the development set for each essay prompt. The scores are the average of three experiments with different seed values.

	# of hidden layers						
Model	1	2	3	4	5	7	10
baseline	.813	–	–	–	–	–	–
cat	.792	**.825**	.814	.813	.801	.766	.722
net	.812	**.824**	.817	–	–	–	–
multi–hum (0.8)	–	.819	**.827**	–	–	–	–
multi–hum (0.6)	–	.804	.812	–	–	–	–
dual–hum (0.8)	–	.816	**.824**	–	–	–	–
dual–hum (0.6)	–	.820	.819	–	–	–	–

Table 2: Comparison of the number of hidden layers in FFNN on the top (type256, Prompt 1, QWK dev)

5 Results

5.1 Hyperparameters for Each Model Architecture

Using `type256` for the grammar features, we searched for the optimal hyperparameters for each model architecture. Table 2 shows the QWK results on the development set of Prompt 1 when we changed the number of hidden layers in the FFNN on the top. When the number of hidden layers was set to 1 in `cat`, QWK was lower than the baseline (.792 vs. .813). QWK became the highest when the number of hidden layers was set to 2, while it got lower as the number of hidden layers increased. In `net`, the architecture with 2 hidden layers achieved the highest QWK. In both `multi-hum`

[6] We tried models with task-specific layers, but the performance was worse than ones without them.

[7] Conventions for Prompt 1, 2, 7, and 8. Language for Prompt 3-6.

[8] https://github.com/huggingface/transformers

	Prompt								
Model	1	2	3	4	5	6	7	8	avg.
baseline	.799	.662	.662	.804	.801	.809	.821	.726	.760
+ type256									
cat	.819	.674	.675	.801	.809	.809	.830	.721	.767
net	.814	.679	.678	.810	.806	.806	.831	.737	.770
multi–hum	.816	.678	.683	.812	.810	.811	.830	.746	.773
dual–hum	.818	.673	.687	.819	.807	.813	.833	.750	**.775**

Table 3: Comparison among model architectures (type256, QWK test)

and `dual-hum`[9], QWK became the highest when the number of hidden layers was set to 3 and the weight of the loss for the main task to 0.8. In the subsequent experiments, we trained models using these hyperparameters.

5.2 Comparison among Model Architectures

Using `type256` for the grammar features, we compared the model performance among the four model architectures. Table 3 shows the QWK results on the test set of all prompts. By using `type256`, the average QWK score for all essays improved in all proposed models, compared to the baseline (See avg. in Table 3).

In `cat`, however, the QWK scores did not improve in three prompts (Prompt 4, 6, and 8), which suggests that simple concatenation of essay representations and grammatical features was not sufficient enough to take advantage of the information that the grammatical features have. In `net`, only Prompt 6 did not improve from the baseline, and it seems effective to feed the grammatical features to an FFNN before concatenating with essay representations.

The QWK scores for the models with the auxiliary task (`multi-hum` and `dual-hum`) were higher than the others. Even when looking at the QWK scores for each essay prompt individually, the scores improved for all prompts. These results suggest that multi-task learning with grammar scores is effective to take advantage of grammatical features.

`Dual-hum` achieved the best performance among the four proposed architectures. In `dual-hum`, grammar scores were predicted from the final layer of the FFNN for grammatical features (see Figure 1), which might let the model learn better representations for grammatical features.

Since the `dual-hum` model performed the best, we conducted the subsequent experiments using the setting.

5.3 Comparison of Grammatical Features

Using the `dual-hum` setting, we compared the effectiveness of different grammatical features. Table 4 shows the QWK results on the test sets when we trained models using different grammatical features.[10]

PFs and NFs In the previous section, we showed that positive linguistic features (PFs; `type256`) improved the AES performance. From the Table 4, we can see that negative linguistic features (NFs; `err54`) also improved the model performance (see avg.). Even on a per-prompt basis, the QWK scores were higher for all prompts than those in the baseline.

Combining the PFs and the NFs (`type256 + err54`) also resulted in an improvement in AES performance. However, the average QWK score (.775) was almost same as that for `type256` and `err54`, and no synergistic effect was observed by using both the PFs and the NFs. We just concatenated the vectors of the two features before feeding the features to the FFNN for grammatical features, and there might be more effective ways to combine them.

PFs weighted using IRT parameters We further explored the effectiveness of PFs by weighting them using IRT parameters (see Section 3.1.2). When we considered the difficulties of individual grammatical items (`multiply_b`), the QWK score became the highest among all settings. On the other hand, modifying `type256` with the probability that a writer with a certain grammar ability uses the grammatical item correctly did not help to improve AES performance. Although the QWK scores got higher than that for the baseline, they were lower than that for `type256`. The results suggest that it is more important to capture what items the writer actually used in the essay than what items the writer seemed able to use.

Effect of Grammatical Features To verify that the score improvement came from the addition of grammatical features rather than multi-task learning, we trained models with the auxiliary task but without using grammatical features. The number of hidden layers in the FFNN on the top

[9] “–hum” represents that grammar scores assigned by human raters were used. “–irt” is added when grammar abilities estimated using IRT are used.

[10] The QWK results for the auxiliary task are shown in Appendix A.

Features	Prompt 1	 2	 3	 4	 5	 6	 7	 8	 avg.
baseline	.799	.662	.662	.804	.801	.809	.821	.726	.760
multi–ffnn1	.803	.680	.659	.797	.802	.806	.827	.723	.762
multi–ffnn3	.812	.671	.684	.812	.805	.812	.831	.748	.772
type256	.818	.673	.687	.819	.807	.813	.833	.750	.775
	(**+.019**)	(+.011)	(**+.025**)	(+.015)	(+.006)	(+.004)	(**+.012**)	(**+.024**)	(+.015)
err54	.815	.672	.689	.813	.805	.812	.832	.756	.774
	(+.016)	(+.010)	(**+.027**)	(+.009)	(+.004)	(+.003)	(+.011)	(**+.030**)	(+.014)
type256+err54	.821	.673	.689	.815	.810	.805	.834	.752	.775
	(**+.022**)	(+.011)	(**+.027**)	(+.011)	(+.009)	(-.004)	(**+.013**)	(**+.026**)	(+.015)
multiply_b	.811	.680	.701	.818	.813	.821	.829	.759	**.779**
	(+.012)	(+.018)	(**+.039**)	(+.014)	(+.012)	(+.012)	(+.008)	(**+.033**)	(+.019)
prob	.820	.661	.682	.813	.807	.808	.834	.752	.772
	(**+.021**)	(-.001)	(**+.020**)	(+.009)	(+.006)	(-.001)	(**+.013**)	(**+.026**)	(+.012)
multiply_prob	.826	.662	.678	.815	.813	.809	.827	.746	.772
	(**+.027**)	(±0)	(+.016)	(+.011)	(+.012)	(±0)	(+.006)	(**+.020**)	(+.012)
add_prob	.812	.674	.682	.806	.799	.812	.827	.757	.771
	(+.013)	(+.012)	(**+.020**)	(+.002)	(-.002)	(+.003)	(+.006)	(**+.031**)	(+.011)
Yang et al. (2020)	.817	**.719**	.698	.845	**.841**	**.847**	.839	.744	**.794**
	(+.017)	(**+.040**)	(+.019)	(**+.023**)	(**+.038**)	(**+.050**)	(+.004)	(+.019)	(**+.026**)
Cao et al. (2020)	.824	.699	**.726**	**.859**	.822	.828	**.840**	.726	.791
	(-.002)	(+.001)	(+.017)	(**+.037**)	(-.002)	(-.001)	(+.011)	(-.017)	(+.006)
Wang et al. (2022)	**.834**	.716	.714	.812	.813	.836	.839	**.766**	.791

Table 4: Comparison among grammatical features (dual–hum, QWK test). The numbers in parentheses indicate the improvement from the baseline. The numbers in parentheses for Yang et al. (2020) and Cao et al. (2020) are the improvement from their baseline, which is equivalent to ours (`RegressionOnly` and `BERT (individual)`, respectively; n/a for Wang et al. (2022)).

was set to 1 (`multi–ffnn1`; same as the baseline) and 3 (`multi–ffnn3`; the best parameter for `multi–hum`; see Section 5.1). The QWK scores for `multi–ffnn1` and `multi–ffnn3` were higher than that of the baseline, but lower than those of the models with grammatical features (Table 4). The results show that both multi-task learning and grammatical features contributed to improve the model performance. In addition, the significant improvement on `multi–ffnn3` suggests that adding layers on the top of BERT would be effective in multi-task learning.

Scoring examples We show some examples from the fold 2 of Prompt 1 (Table 5). The true scores of the four examples are 10, and are written in roughly the same number of words.

In ID 1382, a relatively wide variety of grammatical items were used (10.18 items per 100 words, while the average for essays with true score of 10 included in the fold 2 test set was 9.86). The model trained using `type256` captured the characteristic and predicted the correct score.

Essay ID	# words	Grammatical items # type	 per 100	Predicted score baseline	 type256
1382	442	45	10.18	9	10
377	480	47	9.79	12	11
104	405	38	9.38	9	8
1097	421	42	9.98	9	8

Table 5: Scoring examples. The true scores of the four examples are 10. Per 100 represents the number of different grammar items used per 100 words.

On the other hand, for ID 377 and 104, the model trained using `type256` assigned lower scores than the baseline because of the limited variety of grammatical items in the essays. Note that the prediction improved in ID 377, while it got worse in ID 104.

In ID 1097, our model did not perform well. Although a relatively wide variety of grammatical items were used, the predicted score was lower than that of the baseline.[11]

[11] See Appendix B for the confusion matrix on all the data points.

Model	Prompt 1	2	3	4	5	6	7	8	avg.
multi-irt	.819	.669	.697	.811	.813	.821	.839	.757	.778
dual-irt	.805	.678	.686	.807	.808	.816	.831	.742	.772

Table 6: QWK results of the models using the IRT ability parameter for the auxiliary task (QWK test)

Comparison with existing models The QWK scores for the state-of-the-art AES models are also shown in Table 4. The average QWK score of our models (the highest at .779) was not as high as those of the existing models. In some prompts, there seemed to be differences in baseline QWK scores between the previous studies and ours, and we made comparisons based on the improvement from each baseline[12].

In Prompt 1, 3, 7, and 8, our proposed models showed a greater improvement in the QWK scores compared to Yang et al. (2020) and Cao et al. (2020). In these four prompts, the scores themselves of our models were also competitive with those of the existing models. Cao et al. (2020) achieved the state-of-the-art results in Prompt 3, 4, and 7, but the improvements from their baselines were relatively small in the other prompts.

However, our proposed methods were less effective for Prompt 2, 4, 5, and 6, which resulted in lower average QWK scores than the existing models. To identify when the proposed methods were effective, we examined the characteristics of the essays, such as the type of essays, the average number of words in essays, the correlation coefficient between holistic scores and the grammar ability parameter θ and between human-annotated grammar scores and θ, and the variance of θ, but none of them could provide a satisfactory explanation. We need further investigation and it might help to improve the performance on the prompts where our methods were less effective.

5.4 Using the IRT Ability Parameter for the Auxiliary Task

In Section 5.2, we demonstrated that `dual-hum` model achieved the best performance among the four proposed architectures. However, the architecture requires grammar scores annotated by human raters. Therefore we employed grammar abilities estimated using IRT, which requires no human-annotated labels, as the teacher signals.

Table 6 shows that `multi-irt` and `dual-irt` models achieved comparable performance to the models that used human-annotated score. In general, analytical scoring is more time-consuming than holistic scoring, and grammar scores, which are one of the analytical scores, are not always available in a dataset. A method that improves AES performance without the additional human-annotated labels has practical value. Another advantage of using IRT for our AES models is that we can provide the characteristics of grammatical items (*i.e.*, discrimination and difficulty) as well as essay scores.

6 Conclusions

This study examined the effectiveness of using grammatical features in AES models. Specifically, we fed two kinds of features: (1) grammatical items that writers used correctly in essays (PFs), and (2) the number of grammatical errors (NFs). We showed that both PFs and NFs improved the model performance, but combining them did not result in further improvement. The experimental results suggest that multi-task learning would be effective to take advantage of the information that the grammatical features have. One of the future directions could be exploring effective ways to combine PFs and NFs to improve the model performance since the way in this study was a simple concatenation of the two vectors (*e.g.*, to learn representations for PFs and NFs in different networks and combine them). Another direction would be to examine the effectiveness of adding our grammatical features in AES using a large language model. It potentially improves the scoring performance in zero- and/or few-shot settings (Mizumoto and Eguchi, 2023). Furthermore, in order to have more interpretable models, it would be beneficial to analyze how much individual grammatical features contribute to model's score prediction. The insights delivered by interpretable models can help practitioners in education.

We also weighted PFs in several ways using IRT parameters and found that considering the difficulties of grammatical items would improve the model performance. In addition, we used the ability parameter θ as teacher signals for the auxiliary task in multi-task learning. Although no human-annotated labels are required to estimate the IRT

[12] We re-implemented R^2 BERT (Yang et al., 2020), but our re-implementation of the model did not achieve as good scores as those reported in their paper. Furthermore, we trained models using grammatical features with the loss combination proposed by them (*i.e.*, regression and ranking loss), which resulted in lower QWK scores than our baseline.

parameters, the model trained with the ability parameter achieved comparable performance to the model trained with grammar scores annotated by human raters. In this study, IRT parameters were estimated based on grammatical items that writers used in their essays. In the future, we will apply IRT to both PFs and NFs to model writers' grammar abilities.

7 Limitations

Our proposed methods showed significant advantage on some essay prompts in the ASAP dataset, while they were less effective on the other prompts. Further investigation is necessary to clarify what kind of essays our proposed methods would be effective to. An analysis of the effectiveness of grammatical features on different prompts will also provide additional insights into the variation of model behavior across different prompts.

There are also some limitations related to the extraction of grammatical features. First, the toolkit provided by the CEFR-J Grammar Profile extracts grammatical items based on sophisticated regular expression patterns, which was written by a linguist. It would be quite challenging to prepare a similar toolkit in other languages. Bannò and Matassoni (2022) let a model predict the frequencies of grammatical errors from essay representations, which can be applicable to PFs, but the approach requires human-annotated labels to train a model. Another approach is to extract grammatical features based on cross-linguistically consistent annotations such as Universal Dependencies. It makes easier to use grammatical features in other languages, while it remains challenging to extract ones related to parts of speech and/or morphological features rather than dependencies (*e.g.*, present perfect in English).

Second, there could be errors in the extraction using regular expressions and the same is true for grammatical error correction. Experiments using grammatical features annotated by humans would help reveal the influence of errors in feature extraction.

Third, our method requires explicitly extracting grammatical features at test time as well as at training time. An alternative would be to develop a multi-task learning framework where a model is trained to reconstruct grammatical features at training time and then run the trained model on unparsed test data (*e.g.*, Andersen et al., 2021).

Acknowledgments

A part of this work was supported by JSPS KAKENHI Grant Numbers JP21H05054 and by JST, the establishment of university fellowships towards the creation of science technology innovation, Grant Number JPMJFS2137.

References

Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. 2016. Automatic text scoring using neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 715–725, Berlin, Germany. Association for Computational Linguistics.

Øistein E Andersen, Zheng Yuan, Rebecca Watson, and Kevin Yet Fong Cheung. 2021. Benefits of alternative evaluation methods for automated essay scoring. In *Proceedings of the 14th International Conference on Educational Data Mining (EDM21)*, pages 856–864.

Itsuki Aomi, Emiko Tsutsumi, Masaki Uto, and Maomi Ueno. 2021. Integration of automated essay scoring models using item response theory. In *Artificial Intelligence in Education: 22nd International Conference, AIED 2021, Utrecht, The Netherlands, June 14–18, 2021, Proceedings, Part II*, pages 54–59.

Stefano Bannò and Marco Matassoni. 2022. Cross-corpora experiments of automatic proficiency assessment and error detection for spoken English. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 82–91, Seattle, Washington. 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.

Jill Burstein, Martin Chodorow, and Claudia Leacock. 2004. Automated essay evaluation: The criterion online writing service. *AI Magazine*, 25(3):27–36.

Yue Cao, Hanqi Jin, Xiaojun Wan, and Zhiwei Yu. 2020. Domain-adaptive neural automated essay scoring. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 1011–1020.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28:41–75.

Council of Europe. 2001. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment*. Cambridge University Press, Cambridge.

Hannah Craighead, Andrew Caines, Paula Buttery, and Helen Yannakoudakis. 2020. Investigating the effect of auxiliary objectives for the automated grading of learner English speech transcriptions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2258–2269, Online. Association for Computational Linguistics.

Ronan Cummins and Marek Rei. 2018. Neural multitask learning in automated assessment. *arXiv*, arXiv: 1801.06830.

Ronan Cummins, Meng Zhang, and Ted Briscoe. 2016. Constrained multi-task learning for automated essay scoring. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–799, Berlin, Germany. Association for Computational Linguistics.

Tirthankar Dasgupta, Abir Naskar, Lipika Dey, and Rupsa Saha. 2018. Augmenting textual qualitative features in deep convolution recurrent neural network for automatic essay scoring. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 93–102, Melbourne, Australia. 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.

Yuning Ding, Marie Bexte, and Andrea Horbach. 2023. Score it all together: A multi-task learning study on automatic scoring of argumentative essays. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13052–13063, Toronto, Canada. Association for Computational Linguistics.

Tim Elks. 2021. Using transfer learning to automatically mark L2 writing texts. In *Proceedings of the Student Research Workshop Associated with RANLP 2021*, pages 51–57, Online. INCOMA Ltd.

Ronald K. Hambleton, Hariharan Swaminathan, and H. Jane Rogers. 1991. *Fundamentals of Item Response Theory*. SAGE Publications, California.

John A Hawkins and Paula Buttery. 2009. Using learner language from corpora to profile levels of proficiency: insights from the english profile programme. In Lynda Taylor and Cyril J Weir, editors, *Language Testing Matters Investigating the Wider Social and Educational Impact of Assessment - Proceedings of the ALTE Cambridge Conference April 2008*, pages 158–175. Cambridge University Press.

John A Hawkins and Luna Filipović. 2012. *Criterial Features in L2 English: Specifying the Reference Levels of the Common European Framework*. Cambridge University Press, Cambridge.

Alex Housen, Folkert Kuiken, and Ineke Vedder, editors. 2012. *Dimensions of L2 Performance and Proficiency: Complexity, Accuracy and Fluency in SLA*. John Benjamins, Amsterdam.

Mohamed A. Hussein, Hesham A. Hassan, and Mohammad Nassef. 2020. A trait-based deep learning automated essay scoring system with adaptive feedback. *International Journal of Advanced Computer Science and Applications*, 11(5):287–293.

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 (APCLC 2018)*, pages 160–165.

Zixuan Ke and Vincent Ng. 2019. Automated essay scoring: A survey of the state of the art. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 33–40.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*.

Folkert Kuiken. 2023. Linguistic complexity in second language acquisition. *Linguistics Vanguard*, 9(s1):83–93.

Frederic M. Lord. 1952. *A theory of test scores (Psychometric Monograph No. 7)*. Psychometric Society, Iowa City.

Frederic M. Lord. 1980. *Applications of Item Response Theory To Practical Testing Problems*. Routledge, New York.

Sandeep Mathias and Pushpak Bhattacharyya. 2018. ASAP++: Enriching the ASAP automated essay grading dataset with essay attribute scores. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Sandeep Mathias and Pushpak Bhattacharyya. 2020. Can neural networks automatically score essay traits? In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 85–91, Seattle, WA, USA → Online. Association for Computational Linguistics.

Farjana Sultana Mim, Naoya Inoue, Paul Reisert, Hiroki Ouchi, and Kentaro Inui. 2019. Unsupervised learning of discourse-aware text representation for essay scoring. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 378–385, Florence, Italy. Association for Computational Linguistics.

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.

Panitan Muangkammuen and Fumiyo Fukumoto. 2020. Multi-task learning for automated essay scoring with sentiment analysis. 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: Student Research Workshop*, pages 116–123, Suzhou, China. Association for Computational Linguistics.

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, Florence, Italy. Association for Computational Linguistics.

Robert Ridley, Liang He, Xin-yu Dai, Shujian Huang, and Jiajun Chen. 2021. Automated cross-prompt scoring of essay traits. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13745–13753.

Takumi Shibata and Masaki Uto. 2022. Analytic automated essay scoring based on deep neural networks integrating multidimensional item response theory. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2917–2926, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

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.

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.

Masaki Uto. 2021. A multidimensional generalized many-facet rasch model for rubric-based performance assessment. *Behaviormetrika*, 48:425–457.

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*, 16(6):983–1000.

Masaki Uto and Masashi Okano. 2021. Learning automated essay scoring models using item-response-theory-based scores to decrease effects of rater biases. *IEEE Transactions on Learning Technologies*, 14(6):763–776.

Masaki Uto, Yikuan Xie, and Maomi Ueno. 2020. Neural automated essay scoring incorporating handcrafted features. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6077–6088, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Sowmya Vajjala. 2018. Automated assessment of Non-Native learner essays: Investigating the role of linguistic features. *International Journal of Artificial Intelligence in Education*, 28(1):79–105.

Yongjie Wang, Chuang Wang, Ruobing Li, and Hui Lin. 2022. On the use of bert for automated essay scoring: Joint learning of multi-scale essay representation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3416–3425, Seattle, United States. Association for Computational Linguistics.

Sara Cushing Weigle. 2002. *Assessing Writing*. Cambridge University Press, Cambridge.

Kate Wolfe-Quintero, Shunji Inagaki, and Hae-Young Kim. 1998. *Second language development in writing : measures of fluency, accuracy, & complexity*. University of Hawai'i Press, Honolulu.

Jin Xue, Xiaoyi Tang, and Liyan Zheng. 2021. A hierarchical bert-based transfer learning approach for multi-dimensional essay scoring. *IEEE Access*, 9:125403–125415.

Kevin P. Yancey, Geoffrey Laflair, Anthony Verardi, and Jill Burstein. 2023. Rating short L2 essays on the CEFR scale with GPT-4. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 576–584, Toronto, Canada. Association for Computational Linguistics.

Ruosong Yang, Jiannong Cao, Zhiyuan Wen, Youzheng Wu, and Xiaodong He. 2020. Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1560–1569, 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.

A QWK Results for the Auxiliary Task

Table 7 shows the QWK score for the auxiliary task (*i.e.*, predicting grammar score). The QWK scores were generally low, and some of them were negative. We observed that the models output scores

	Prompt								
Model	1	2	3	4	5	6	7	8	avg.
type256	0.032	-0.007	0.050	-0.007	0.001	0.000	0.000	0.079	0.016
err54	-0.002	0.017	-0.003	0.014	-0.007	0.003	-0.012	0.045	0.008
type256+err54	0.148	0.003	0.085	0.001	0.000	0.001	-0.002	0.110	0.039
multiply_b	0.015	-0.003	-0.002	-0.023	0.000	0.000	0.012	-0.003	-0.001
prob	0.052	-0.025	0.028	0.000	0.000	0.000	0.000	0.046	0.008
multiply_prob	0.097	0.000	0.003	0.000	0.000	0.000	0.007	0.059	0.018
add_prob	0.053	-0.023	0.050	-0.002	0.004	0.000	0.039	0.004	0.011

Table 7: QWK results for the auxiliary task on the test set (models shown in Table 4)

close to the mode value in each prompt. One of the possible reasons is the relatively low weight for loss function for the auxiliary task (*i.e.*, 0.2). However, when we assigned a higher weight for the auxiliary task (*i.e.*, 0.4), the model prediction for the main task got worse. Further consideration is necessary for predicting multiple essay traits simultaneously (*e.g.*, Ridley et al., 2021; Shibata and Uto, 2022).

B Detailed Results of Model Predictions

Detailed scoring performance of the model trained using `type256` is shown in Figure 2. The values in the confusion matrices are the sum of all experiments (*i.e.*, 5-fold cross validation and three experiments with different seed values).

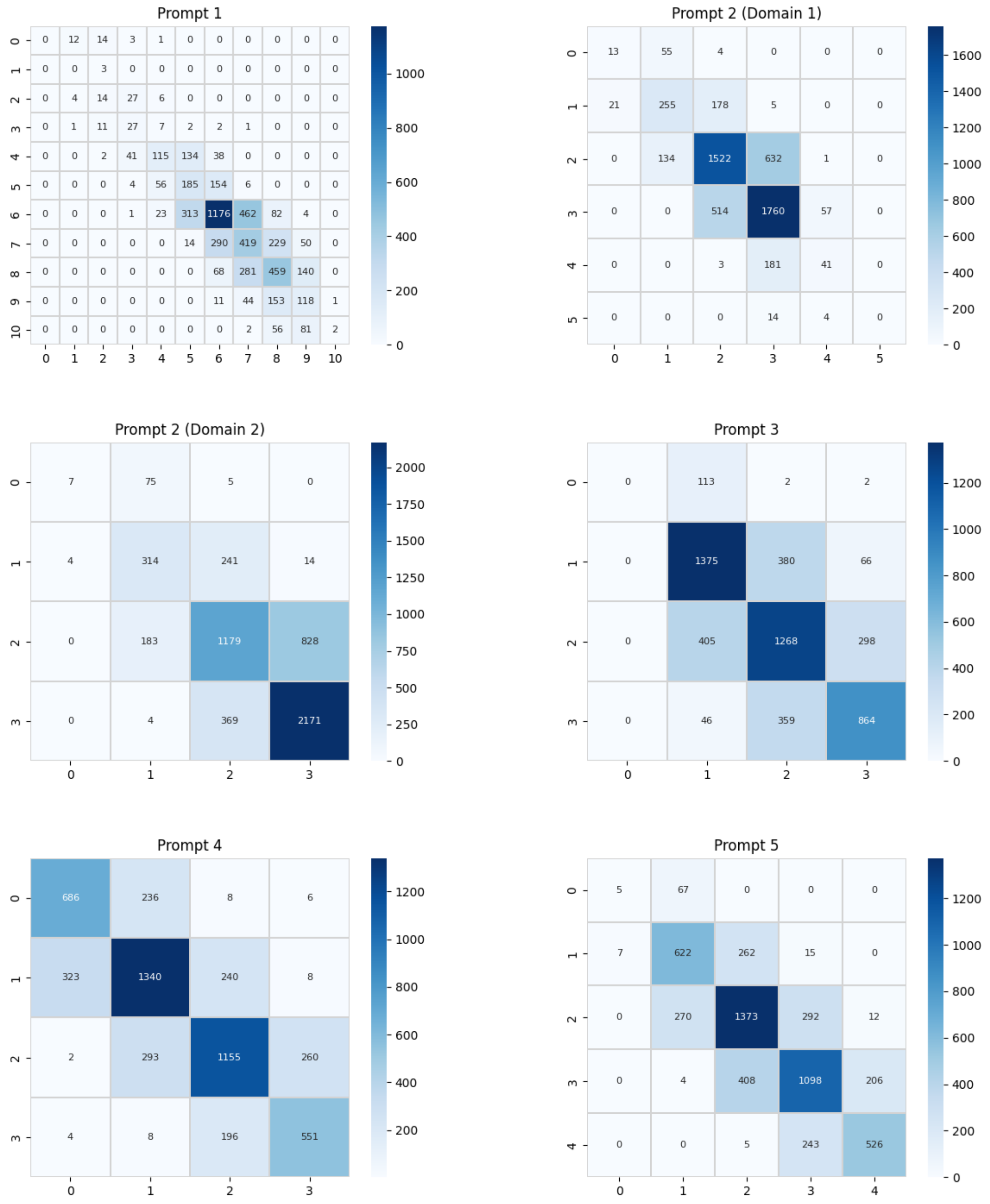
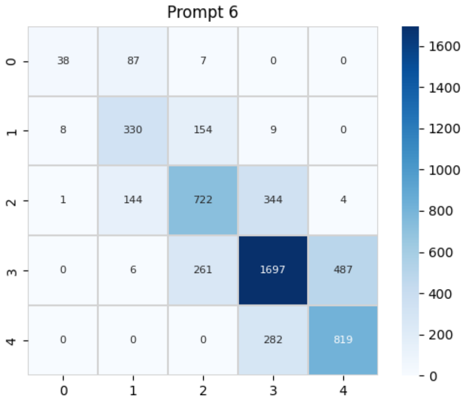
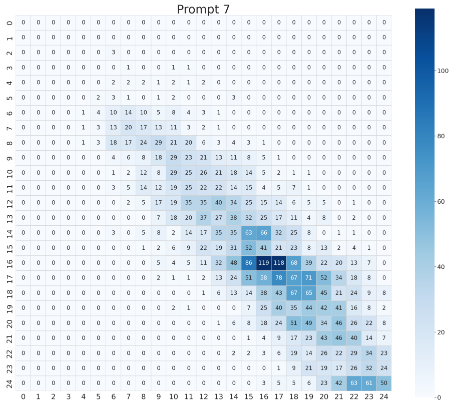


Figure 2: Scoring performance of the model trained using `type256`





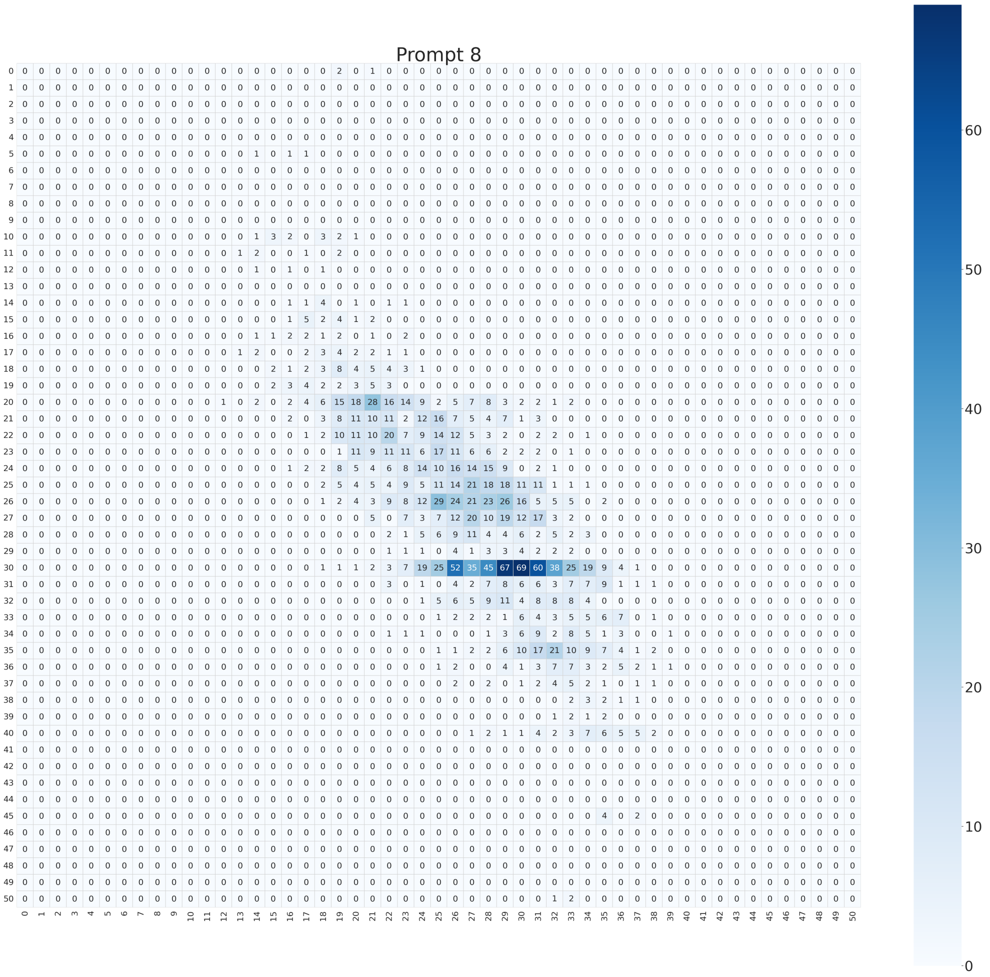


Figure 2: Scoring performance of the model trained using `type256` (*cont.*)

Error Tracing in Programming: A Path to Personalised Feedback

Martha Shaka, Diego Carraro and **Kenneth N. Brown**
Centre for Research Training in AI, Insight, School of Computer Science & IT,
University College Cork
Ireland
m.shaka@cs.ucc.ie, diego.carraro@insight-centre.org, k.brown@cs.ucc.ie

Abstract

Knowledge tracing, the process of estimating students' mastery over concepts from their past performance and predicting future outcomes, often relies on binary pass/fail predictions. This hinders the provision of specific feedback by failing to diagnose precise errors. We present an error-tracing model for learning programming that advances traditional knowledge tracing by employing multi-label classification to forecast exact errors students may generate. Through experiments on a real student dataset, we validate our approach and compare it to two baseline knowledge-tracing methods. We demonstrate an improved ability to predict specific errors, for first attempts and for subsequent attempts at individual problems.

1 Introduction

The increasing importance of digital technologies has made programming a critical skill. The teaching of programming has long been recognised as difficult, and novice programmers often struggle with syntax, and with conceptual and problem-solving skills (Figueiredo and García-Peñalvo, 2021; Thuné and Eckerdal, 2019). Practical assignments, designed to enhance understanding, often become stumbling blocks due to compiler errors that are not informative for beginners, leading to confusion or discouragement (Medeiros et al., 2019). Further, given large class sizes, providing personalised feedback from instructors is difficult (Parihar et al., 2017; Song et al., 2019). Recent research has explored Automatic Feedback generation, including test-case analysis (Xiong et al., 2018) and AI-driven Automatic Program Repair systems (Bhatia and Singh, 2016; Gulwani et al., 2018; Suzuki et al., 2017). But many of these system fail to trace the individual learner's profile or unique learning trajectory, thus reducing the effectiveness of the feedback provided (Ghosh et al., 2021).

In contrast, Knowledge Tracing (KT), an educational data mining technique, has the potential to create personalised learning experiences by predicting student performance based on their mastery of concepts (Piech et al., 2015; Wang et al., 2017; Emerson et al., 2019). In programming education, KT is useful for recommending exercises, predicting assignment outcomes, and identifying students at risk of underperforming (Huang et al., 2019; Azcona et al., 2019). But traditional KT models often overlook the granularity of student responses, treating all correct or incorrect attempts uniformly (Ghosh et al., 2021). Programming errors, though, vary widely, from simple syntax mistakes like a missing semicolon, to more complex issues such as failing to implement a loop correctly. Deep Knowledge Tracing (DKT) (Piech et al., 2015) uses neural networks to identify specific patterns, and thus allows more specific feedback.

This paper propose a refined application of DKT to identify precise compiler errors. By analysing the error patterns in students' historical performance, we aim to identify the specific concepts or syntax elements that a student has not yet mastered. This then enables the delivery of targeted feedback focused on those elements. In addition, by analysing the patterns of multiple students in a class, we can highlight common error patterns, for further action by educators.

Our contributions are as follows. (1) We introduce a novel KT task, error-based knowledge tracing, to learn a meaningful representation of student submissions. We introduce a new error-based deep knowledge tracing model (Error-DKT) to track the progressive student error patterns. (2) We conducted experiments on a real-world student code database and found that incorporating error features significantly enhances the accuracy of specific error predictions, elevating the F1 score from 0.27 (as seen in existing models) to 0.5. (3) We discuss the broader implications and limitations of this

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 330–342
June 20, 2024

research within programming education, proposing new research directions to bridge the gap between generic feedback systems and the need for individualised educational support.

2 Related Work

Knowledge Tracing (KT) is designed to predict students' future performance by analysing their past interactions with learning materials. Initially, KT relied on probabilistic models such as Bayesian Knowledge Tracing (BKT) (Corbett and Anderson, 1994), which estimates students' mastery using a Bayesian Network and a set of fixed parameters (guess, slip, learn, and sometimes forget). BKT's extended by Käser et al. (2017) through the introduction of Dynamic BKT, to account for interactions between different knowledge components.

Deep Knowledge Tracing (DKT) leverages recurrent neural networks to harness the sequential patterns in student interaction data, effectively capturing not only correctness of responses but also the order and context of these interactions (Piech et al., 2015). Recent advances includes techniques such as attention mechanisms (e.g., AKT-Context-aware attentive knowledge tracing (Ghosh et al., 2020)), external memory modules (e.g., DKVMN-Dynamic key-value memory networks for knowledge tracing (Zhang et al., 2017)), and GKT-Graph-based KT (Nakagawa et al., 2019), each aiming to better understand the learning process's complexities. DKT has outperformed recent deep learning models (Shi et al., 2022; Liu et al., 2022). Liu et al. (2023); Abdelrahman et al. (2023) gives a comprehensive review of KT models.

Traditional DKT models primarily rely on sequences of question numbers and the correctness of attempts for prediction, often overlooking detailed information about students' approaches to solving questions (Shi et al., 2022; Ghosh et al., 2021; Abdelrahman et al., 2023). This omission restricts their predictive power across different domains. However, incorporating domain-specific features has been shown to enhance performance. For example, in the mathematical domain, Liu et al. (2020) enhanced predictions by including question-concept relationships derived from Pre-training Embeddings via Bipartite Graph (PEBG), while in the programming domain, Shi et al. (2022) introduced code features using code2vec.

There has been a push to extend DKT's application beyond mere correctness prediction. Ghosh et al. (2021) adapts DKT to forecast the specific options students select in multiple-choice questions. Inspired by this, our work aims to tackle the more complex scenario of open-ended programming questions, which creates the challenge of interpreting diverse compiler errors. Liu et al. (2022) develops Open-ended Knowledge Tracing (OKT), which integrates an enhanced DKT model with code features from an abstract syntax tree neural network-ASTNN (Zhang et al., 2019) and textual question features from GPT-2, aiming to predict student performance. They then employ a GPT-2-based text-to-code generator, guided by the DKT model's hidden state as a knowledge estimate, to generate diverse code solutions that mirror the student's comprehension.

3 Methodology

Our work introduces an alternative approach for DKT to predict directly the specific errors students are likely to encounter. We assess how different domain features like student code submissions, reference solutions, and question-concept relationships affects error prediction. After pinpointing individual errors, we employ a bottom-up approach, aggregating these error predictions to assess overall student performance as pass (error-free) or fail (submission with errors). This is to assess whether focusing on granular error predictions can enhance the accuracy of student outcome forecasts compared to traditional DKT predictions. Figure 1 illustrates our proposed model structure.

3.1 Dataset

We use a dataset from a US university's Spring Semester introductory Java programming course, conforming to ProgSnap2 format (Price et al., 2020). This dataset includes data from 410 students across five assignments, totaling 50 programming questions that assess various concepts like loops and conditions. Students submitted multiple attempts per exercises until achieving a 100% score, with submissions ranging from 10 to 20 lines of code and automatically graded based on test cases.

We focused on Assignment 1, which consists of 10 questions, selected for its high error frequency and variety, providing a comprehensive base for error analysis (details in Table 1). Our analysis employs two subsets: **Set-I**, categorising submissions with compiler errors as *"incorrect"* and those with-

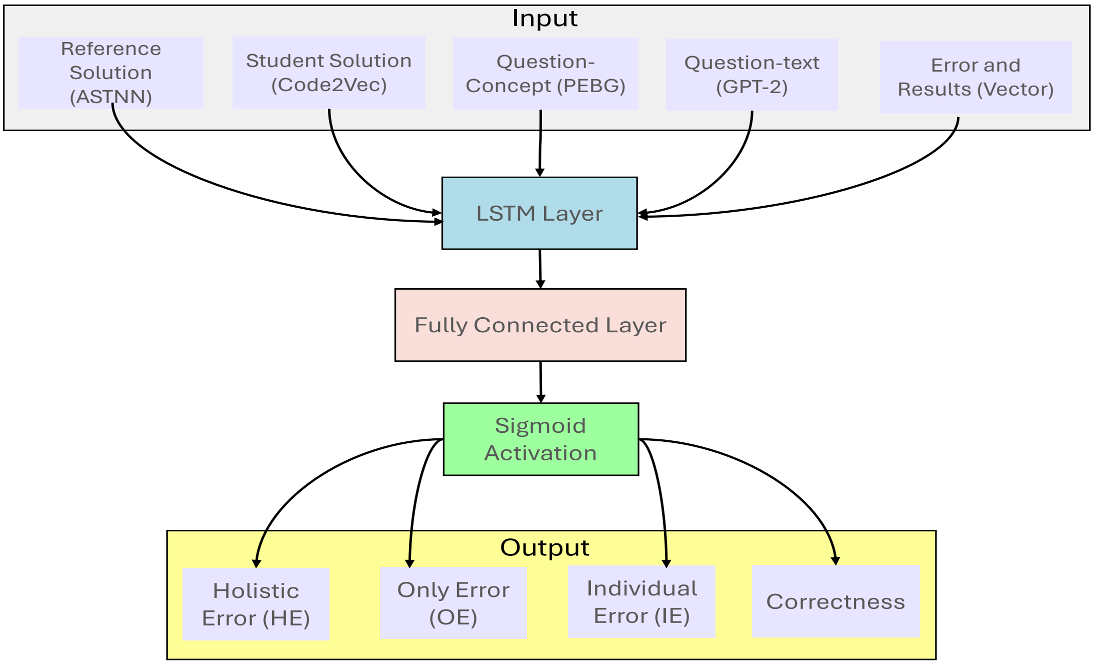


Figure 1: Architecture of Error Tracing, integrating various feature embeddings and LSTM layers to predict student errors and overall performance, as described in section 3.3.

Description	Dataset
Total submissions (subs)	9995
Subs with errors	5948 (59.5%)
Avg errors per subs	1.6
Top 3 frequent errors	[0, 1, 5]
Top 2 common pairs	[1, 3] [1, 2]
Total No of students	386
Avg students per question	368
Most attempted question	5 (≈ 4000 errors)
Least attempted question	4 (≈ 750 errors)

Table 1: Key Features of the Dataset: Summarises submission counts, error rates, common errors, and student engagement metrics, highlighting critical areas of focus within student interactions.

ID	Description	Frequency
0	Passed/ No error	4047
1	’ID’ expected e.g like ";)("	2128
2	Missing return statement	1291
3	Illegal start of expression	1163
4	not a statement	850
5	’else’ without ’if’	629
6	Cannot find symbol: variable ID	624
7	Bad operand types for binary operator ’ID’, like "&&, ‖,*,+,>=,<"	554
8	Incompatible types, like datatypes mismatch	444
9	Reached end of file while parsing, maybe a missing delimiter or closing brace	426

Table 2: Overview of key error types in student submissions, presenting both the frequency and characteristics highlighting common obstacles in the learning process.

out as *"correct"*, specifically for error prediction. **Set-II** is for binary (pass/fail) prediciton, labelling any submission without a perfect score as *"incorrect"* due to compiler or logical errors, and those with full marks as *"correct"*.

To mitigate class imbalance in Set-I, we identified the top 10 errors for proof of concept which includes nine error types and a pass class, with occurrences from 5000 to 400 across the questions, detailed in Table 2.

3.2 Problem Definition

Our approach treats students’ code submissions as a temporal sequence, aiming to trace their concept mastery over time. Each submission at time step t is represented as $x_t = \langle p_t, c_t, s_t, e_t, r_t, \{ref\}_t \rangle$, encapsulating the problem p_t, concept c_t, code solution s_t, errors e_t, result (pass/fail) r_t, and reference

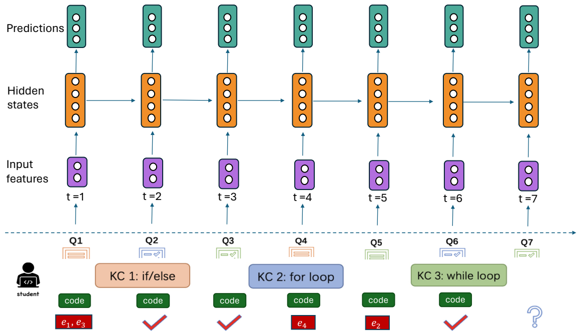


Figure 2: Overview of a simplified RNN Model. The model performs predictions at each timestep, using the previous hidden state (representing estimated mastery) complemented by a diverse input features.

solution $\{ref\}_t$. Given T is the maximum number of attempts, we define students submission as S_T = $\{x_1, x_2, x_3, \ldots, x_T\}$. Our aim is to predict the specific errors e_{T+1} that might arise in the next problem p_{T+1}, based on the student's previous submissions. For example, as illustrated in Figure 2, our target is identifying potential errors (e_7) at time step t_7 based on submissions from t_1 to t_6, while correctness prediction determines the likelihood of a pass/fail result (r_7).

3.3 Error Tracing with DKT

We built an error tracing model (Error-DKT [1]) that utilises a Long Short-Term Memory (LSTM) neural network and a combination of prediction strategies to solve the challenges in multi-label error prediction. As illustrated in Figure 1, it includes constructing detailed input features and a layered architecture, featuring an LSTM layer to discern hidden knowledge states, and a fully connected layer that converts LSTM outputs for multi-label prediction. We investigated two predictive strategies:

Standalone prediction: This strategy employs methods where individual models operate independently to make error predictions.

Holistic Error prediction (HE) a single model is trained to identify probabilities for specific error classes, including a unique "no error" class. This model employs a dynamically adjustable threshold to determine the overall presence/absence status. For instance, if there are four possible error types, the HE model will predict among five outcomes, where one represents the absence of errors

Only Error prediction (OE) focuses solely on detecting errors in a submission. Referring to the example above, OE model will predict the presence of four error classes; if all predictions fall below a certain threshold, the submission is classified as error-free.

Individual Error prediction (IE) a separate model is trained for each error type. Using the same example with four error types, four distinct models would be trained. Their predictions are then aggregated to formulate a comprehensive view of the errors in a student's attempt.

Ensemble Methodology: This two-step approach initially evaluates the likelihood of any error occurrence before pinpointing exact errors using insights gained from the initial assessment. The Ensembled Error Prediction strategy combines the strengths of conventional DKT in determining submission correctness with the detailed error tracing capabilities of our model to isolate precise errors.

3.4 Baseline Models

To tackle the novel challenge of predicting specific programming errors without established benchmarks, we develop two baseline models using statistical probabilities. The **Simple Baseline Model** use overall dataset statistics to forecast error probabilities, identifying the two most frequent errors per question from historical data. In contrast, the **Complex Baseline Model** offers a granular analysis, calculating error probabilities for each question-attempt pair and pinpointing the two most common errors based on historical data, though it overlooks individual error histories. Additionally, we benchmark against the Open-ended Knowledge Tracing **OKT** approach (Liu et al., 2022), which employs a large language model to generate student code. We run that code through a compiler to identify expected errors, excluding the errors not included in our set thus providing a direct comparison with our error tracing model.

4 Experimental Setup

Our experimental setup, detailed below, outlines the data collection methods, model training protocols, and evaluation metrics used to rigorously test the efficacy of our proposed models.

4.1 Data Preprocessing

We grouped the submissions by student and divided them into training and test sets with a ratio of 4:1. A random split method is used for performance prediction, with an iterative stratification

[1] https://github.com/kencyshaka/Error-DKT

technique, specifically *MultilabelStratifiedShuffleSplit* (Sechidis et al., 2011), used to address class label imbalances for error prediction. We further split the training set to allocate 25% for validation, facilitating hyperparameter tuning. The entire training dataset, including the validation subset, was subsequently utilised for model training, with performance evaluation conducted on the test set.

4.1.1 Constructing Input Features

The input feature x_t for each timestep is:

$$\begin{aligned} x_t =& [\mathrm{E_r}(r_t) \oplus (\mathrm{E_p}(p_t) \odot \square) \oplus (\mathrm{E_c}(c_t) \odot \square) \\ & \oplus (\mathrm{E_{ref}}\{ref\}_t \odot \square) \oplus (\mathrm{E_{er}}(\{er\}_t) \odot \square)] \end{aligned} \quad (1)$$

$\odot$ and $\square$ signify element-wise multiplication and the binary presence or absence of embeddings, respectively. $\oplus$ concatenates to create the final embedding, integrating the problem content ($\mathrm{E_p}$), student and reference code ($\mathrm{E_c}$ and $\mathrm{E_{ref}}$), and errors ($\mathrm{E_{er}}$), alongside results ($\mathrm{E_r}$) to effectively predict student performance.

Problem and Code Embeddings Problem representation (E_p) merges textual content (E_{p1}) and concept relationships (E_{p2}) into a comprehensive embedding. E_{p1} leverages a GPT-2 model trained on Java datasets for textual transformation (Liu et al., 2022), while E_{p2} employs a bipartite graph to capture problem-concept dynamics, following the PEBG methodology (Liu et al., 2020).

Code representation adopts ASTNN (Zhang et al., 2019) for the reference solution (E_{ref}) and a modified code2vec (Alon et al., 2019) approach for student submissions (E_c), facilitating dynamic adaptation during model training (Shi et al., 2022).

Categorical Embeddings Categorical features, such as error lists and outcome indicators, are transformed into vector representations. Error lists are encoded into binary vectors (E_{er}), with the vector size reflecting the total number of distinct errors. Similarly, result embeddings (E_r) denote attempt results and question interactions (Piech et al., 2015), utilising a binary format to represent the data efficiently.

4.2 Network Architectures and Hyperparameter Optimisation

We systematically explored hyperparameters to identify the optimal model configuration, assessing their impact on model performance through average loss and F1 scores on the validation dataset. This iterative process, conducted 100 times, aimed to pinpoint the hyperparameter set yielding the best validation results, which was then applied across the entire training set to construct the final model for subsequent testing and evaluation phases.

Input features, including code embeddings ($\mathrm{E_c}$), reference solution embeddings ($\mathrm{E}_{\{ref\}}$), and textual problem embeddings ($\mathrm{E_{p1}}$), were configured following default parameters from prior studies in Code-DKT (Shi et al., 2022) and OKT (Liu et al., 2022). For the problem-concept relationship component ($\mathrm{E_{p2}}$), we use the PEBG framework, varying parameters such as embedding size ($d = \{\mathbf{64}, 128\}$), epochs ($10, 50, \mathbf{100}, 200$), learning rate ($\mathbf{0.001}, 0.005, 0.0015$), hidden states ($\mathbf{128}, 256$), and batch size ($\mathbf{16}, 32, 128$), with the optimal settings highlighted in bold.

Our architecture exploration was tailored to specific tasks, employing varying hyperparameters to refine the model's structure. This included adjustments to LSTM layers ($1, 2, 4, 8, 10$), learning rates (uniform distribution, min=0.00001, max=0.001), batch sizes (16, 32, 64, 128), epochs (10, 20, 40, 50, 70, 100), threshold settings, and loss types (Binary Cross Entropy, Focal Loss (Lin et al., 2018), Class Balanced (Cui et al., 2019) and Distributed Balanced Loss (Wu et al., 2020)). The selected hyperparameters for each multi-label task in Section 3.3 are summarised in Appendix A.3 Table 6.

Model training and evaluation on an NVIDIA A-40 GPU averaged 10 minutes, while the same tasks took about 4 hours on a local CPU. For further details, see Appendix A.3, Table 6. We use the Adam optimizer for learning rate scheduling in training. Consistent with prior research (Shi et al., 2022), we limited the number of student attempts to 50 for each problem, focusing on the most recent submissions to better reflect current understanding and skills.

4.3 Evaluation Metrics

Model performance: The primary metric for error prediction is the weighted average F1 score, tailored to reflect the proportion of each error class within the dataset. This approach guarantees a balanced evaluation, highlighting the model's precision for common errors while proportionally considering less frequent ones. Weighted average precision and recall further detail the model's predictive accuracy. Additionally, we use the weighted average F-beta score, emphasising precision more than recall. This prioritisation is crucial, as it ensures that any predicted errors intended to guide interventions are reliably identified, maximising the relevance and efficacy of educational support. For performance prediction on the correctness (pass/fail), we use the Area Under the Receiver Operating Characteristic curve (AUC) alongside the average F1 score to assess model performance.

Educational Context: We analyse the model's performance in two educational scenarios: overall accuracy and accuracy in predicting the first attempt at solving a problem. The latter is crucial for identifying early intervention opportunities in knowledge tracing (Emerson et al., 2019), while the overall performance metric helps differentiate between types of errors (conceptual vs. syntactical) and debugging skills.

Problem and Error Analysis: Further, we evaluate the model's effectiveness across individual questions to capture how well historical performance data informs future error predictions. We also evaluate the model performance on the most common to the least frequent errors. This analysis is crucial for understanding the model's capacity to predict common errors (easy task) and uncommon errors (hard task).

Model	**First**		**Overall**	
	F1	F-beta	F1	F-beta
Simple	30.5	32.6	22.9	21.1
Complex	40.9	39.7	26.1	**25.1**
OKT	**47.1**	**41.8**	**27.5**	22.9
HE-BCE	49.2	48.8	42.8	**42.5**
HE-FL	**50.8**	**48.2**	**43.3**	41.8
OE-BCE	20.2	19.8	16.7	17.3
OE-DB	16.5	19.3	30.7	30.3
IE	17.0	19.4	34.1	34.1
HE-BCE	52.0	51.9	43.7	44.8
HE-FL	**53.1**	**53.1**	**50.7**	**50.1**
OE-BCE	52.2	51.2	36.4	37.2
OE-BCE*	52.6	51.6	44.7	44.4
OE-DB	51.5	51.4	45.2	45.7
IE	51.4	51.6	46.9	47.8

Table 3: Evaluation of Model Performance Across Error Prediction Tasks: The table presents F1 and F-beta scores for 'First' and 'Overall' attempts. It starts with baseline model metrics, progresses through Error-DKT error prediction tasks (Holistic Error), OE (Only Error), IE (Individual Error), and concludes with ensemble approaches combining Error-DKT predictions with DKT outcomes. OE-BCE* denotes the model trained solely on submissions with errors. The losses are BCE-Binary Cross Entropy, FL-Focal and DB-Distributed Balance. Bold values highlight top performance within each section.

5 Results

Results are shown in Table 3, for the baselines, the error prediction tasks and the ensemble approaches.

5.1 Error Prediction

Predictive Performance The Error-DKT models, employing single-step and ensemble strategies, outperform baselines, e.g, OKT by +15.8% and +23.2% respectively, showcasing their superior performance in predicting overall student errors. This efficacy is particularly highlighted in the ensemble approach, which underscores the benefit of first identifying error-free submissions before employing Error-DKT models to pinpoint specific student errors, thereby improving overall prediction accuracy. Specifically, focused error prediction models (OE and IE) benefit from this approach, e.g, OE using Distributed Balance loss has +35% increase in accuracy for first attempts.

Performance at predicting first attempt is generally higher than for all attempts, including for OKT (Liu et al., 2022). We believe this is because each question is initially the same for each student, and cohort data for previous questions is informative. Once a student has submitted an attempt that fails, the student is then responding to the compiler messages, and so the task becomes individualised, negating the benefit of more data on each individual student.

Interestingly, the holistic approach (HE) demonstrated superior performance over focused error predictions (IE, OE), especially in contexts with limited data and high imbalance, indicating the challenges of granular error prediction. The enhanced performance of IE and OE in overall attempts, as opposed to first attempts, suggests that accumulating more data leads to improved accuracy. Furthermore, utilising various loss functions to tackle class imbalances significantly enhances model performance. For instance, employing Focal loss results in a +1.6 improvement for HE prediction compared to Binary Cross Entropy, and dynamically adjusting thresholds for error classes also contributes to this advancement.

Per Problem The analysis shows variations in predictive model performance, which could be due to the distinct challenges, skill requirements and

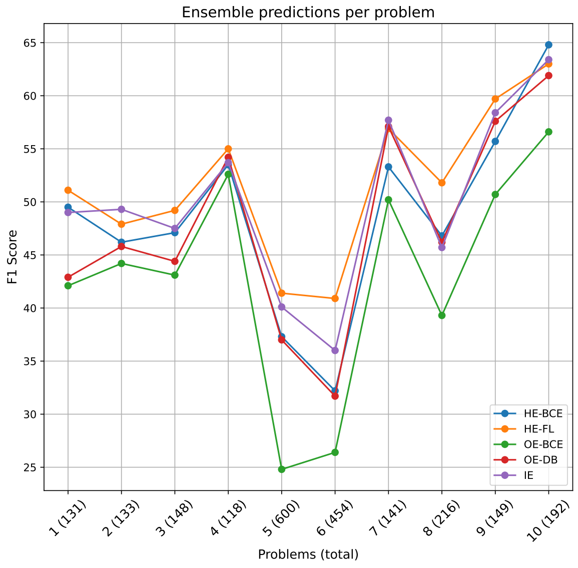


Figure 3: Ensemble Model's Performance per Problem: Overall Attempts.

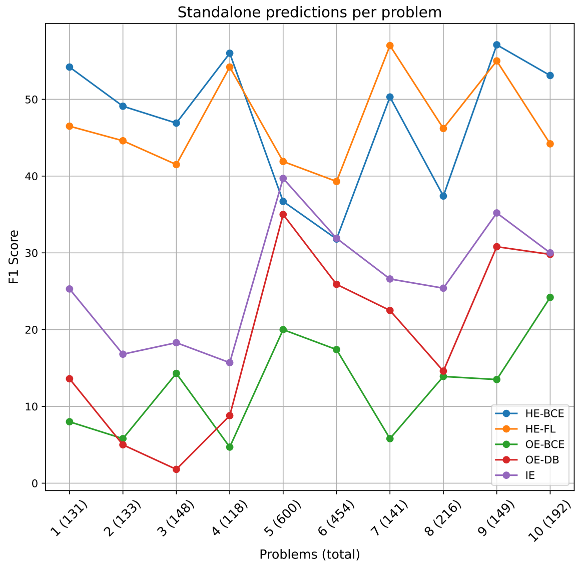


Figure 4: Error-DKT Model's Performance with various Prediction tasks per Problem: Overall Attempts.

prevalence of errors in each problem. A general pattern shows that the model's performance increases in predicting student errors as they advance in their assignments, as shown in Figure 3. This trend emphasises the crucial role of historical performance data in enhancing error prediction for Error-DKT.

Also, we observe the volume of submissions and the frequency of errors committed by students, which emerge as significant factors influencing model predictions due to the diverse and personalised strategies students employ. For example, problem 5 and 6 exhibits a significant decline in prediction accuracy, as highlighted in Figure 3 primarily due to their high error rates—twice and three times more than other problems, respectively (see Appendix A.2 Figure 10). In addition, by comparing the student-problem attempts in Figure 9 and Table 5 in Appendix A.2, we can see that problem 5 and 6 require many more attempts per student, and so appear to be different from the other questions. More attempts means we are again predicting the response to the compiler messages, and our predictive performance declines

In contrast, as shown in Figure 4, focused error prediction models (OE, IE) benefit from more error data, enabling these models to fine-tune their predictions more effectively compared to the damage they cause to the holistic model (HE). Furthermore, the analysis reveals that models perform better on problems that require similar skill sets in later stages (e.g., Problems 7, 9, and 10), suggesting that Error-DKT can successfully model students' knowledge of common error patterns.

Per Error Figure 5 shows the model struggling to accurately predict rare errors across different error classes. Nevertheless, an uptick in the models' ability to predict errors in classes 3 and 7, likely due to their frequent occurrence in problems 5 and 6 (see Appendix. A.1, Figure 7), suggests models like IE can benefit more. Additionally, errors 4 and 5, less common but often occurring with common error 1 (see Appendix. A.1, Figure 8), exhibit enhanced prediction accuracy. This indicates that models successfully extract insights from prevailing error patterns, thereby improving their predictive capabilities. We also note that the OKT models predominantly predicted the error class 2, "missing a return statement". This observation suggests that the estimated student knowledge level failed to prompt the LLM to generate codes incorporating previously unseen errors, such as those involving missing semicolons or unclosed curly brackets.

5.2 Correctness Prediction

Our methods focus on predicting individual errors, raising the question of whether these predictions can be aggregated into a holistic pass/fail assessment. According to the results in Table 4, this approach yields poorer performance compared to the original DKT method, which directly evaluates pass/fail outcomes. However, by incorporating the diverse input features outlined in Equation 1, we can significantly improve the correctness prediction

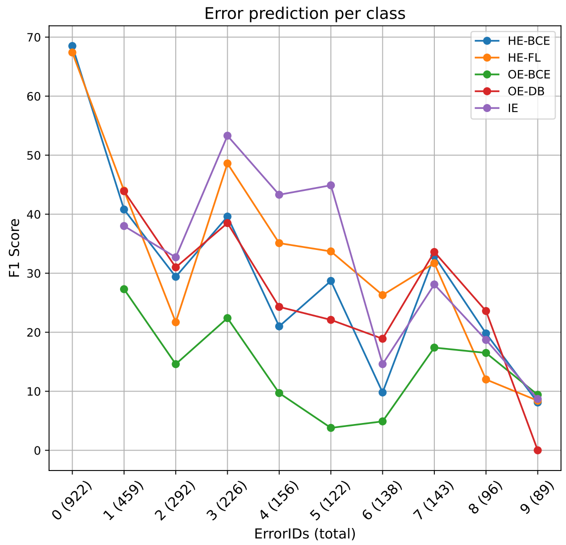


Figure 5: Model's Performance with various Prediction tasks per Error class.

capabilities of the original DKT model.

Model	**First**		**Overall**	
	AUC	F-beta	AUC	F-beta
Simple	46.7	40.9	50.9	58.1
Complex	58.0	46.0	61.6	67.2
OKT	n/a	30.8	n/a	43.0
HE-BCE	73.0	66.3	72.5	77.5
HE-FL	71.9	67.1	65.5	74.3
OE-BCE	68.0	60.5	63.4	70.1
OE-DB	68.7	60.8	68.7	75.1
IE	68.4	60.5	65.3	71.3
DKT	75.5	**72.7**	75.3	78.5
DKT*	**76.9**	72.3	**77.4**	**79.1**

Table 4: Model performance (AUC, F-beta) evaluation based on correctness (pass/fail) prediction. DKT* is trained using the new set of input features

5.3 Knowledge-driven prediction of students' submissions

The heatmaps presented in Figure 6 illustrate the capabilities and limitations of the Error-DKT model. The model exhibits proficiency in predicting errors that occur frequently but shows difficulty in identifying rarer errors. The effectiveness of the ensembled (two-step) approach is evident, as the accuracy of Step I predictions directly influences the subsequent error identification. For example, in Case 1, despite Step I yielding false positives, Step II strongly indicates the presence of errors, which are confirmed with ground truth values. This suggests the potential for alternative ensemble strategies that might allow Step II predictions to carry more weight. In contrast, Case 2 highlights that enhancing the accuracy of Step I predictions, which is generally more straightforward, could potentially lead to overall better performance in the model.

6 Conclusion and Future Works

In our study, we enhanced traditional knowledge tracing methods by developing a framework capable of predicting overall correctness and specific student errors. Our Error-DKT models demonstrated significant effectiveness, substantially outperforming baseline OKT models in overall attempts prediction with improvements of +15.8% and +23.2% using single-step and ensemble strategies (Holistic Error prediction), respectively. The ensemble approach significantly enhances accuracy by initially distinguishing error-free submissions from erroneous ones, and then specifically pinpointing the errors in submissions forecasted to fail.

Predictions for initial attempts generally exhibit higher accuracy, likely due to the uniformity of these submissions and the rich historical data available. However, as students revise their submissions in response to compiler feedback, the complexity of prediction increases, particularly for subsequent attempts. This issue is compounded in problems with high error rates and frequent submissions, like Problems 5 and 6, where performance notably declines. Despite the advantages in error prediction, our method showed less effectiveness in integrating individual errors into holistic pass/fail assessments compared to direct evaluations by traditional DKT methods. Nonetheless, the integration of diverse input features enhances the DKT model's ability to predict correctness. These findings underscore the potential of Error-DKT to improve the precision of error predictions and affirm the ongoing need for models that can adapt to complex error patterns and improve feedback mechanisms in educational settings.

For future work, several promising directions emerge. First, experimenting with advanced DKT architectures like AKT and DKVMN and refined ensemble methods may improve error prediction and accommodate a wider array of error types with more extensive datasets. Secondly, optimising the OKT model to extend its predictive competence to logical as well as compiler errors could yield more comprehensive error detection. Thirdly, there's

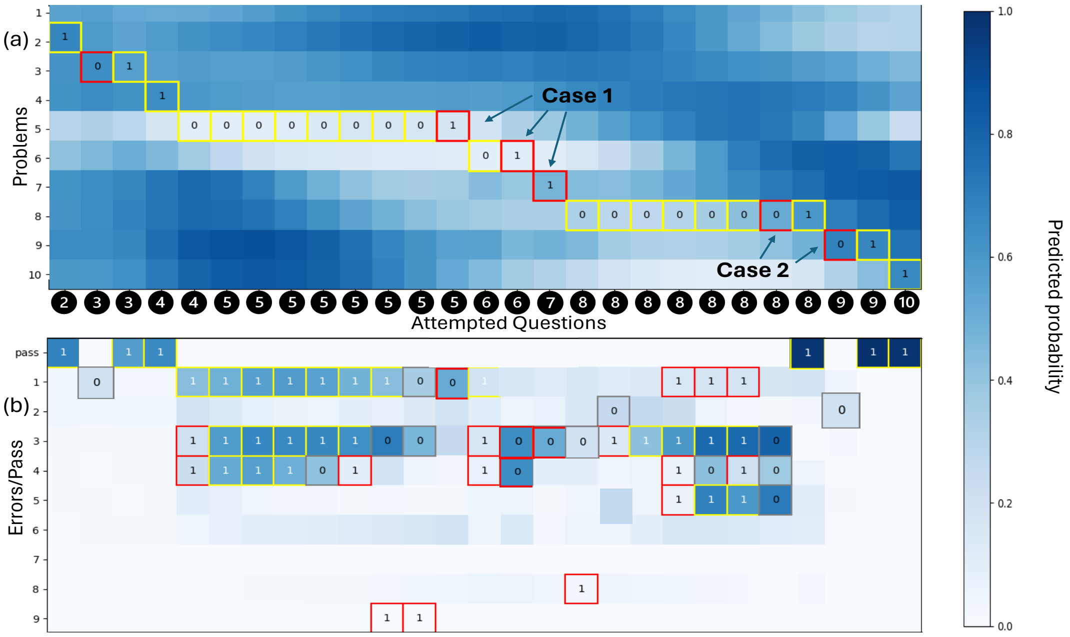


Figure 6: Ensembled Error-DKT prediction heatmap for a student over 27 attempts: (a) showcases Step I's correctness predictions, with yellow boxes indicating accurate predictions and red boxes highlighting incorrect predictions and the numbers in the cells represent the ground truth values, (b) displays specific error predictions using the Holistic approach, where red boxes with '1' signify undetected errors, and those with '0' indicate errors incorrectly predicted absent due to Step I's assessment. Grey boxes represent false error predictions.

a significant opportunity to enhance knowledge tracing models to interpret learned patterns, correlating them to specific knowledge areas, such as debugging skills reflected in students' coding progression. Finally, integrating our framework with an automated feedback mechanism will be vital in evaluating its effectiveness in delivering personalised, actionable feedback to students.

7 Limitations

Our study faces certain limitations. First, the modest performance of our Error-DKT models is partly due to the challenging prediction task and a small dataset (386 student summaries, referenced in Table 1). Despite this, Error-DKT shows promise in identifying specific student struggles better than baseline models. Second, we focus on a narrow dataset from one assignment and semester, limiting generalisation to wider programming contexts or error types. Given the novelty of this KT task, our concentration was solely on predicting compiler errors, with no examination of logical errors. This scope raises questions about the model's applicability across various programming scenarios. Lastly, using only DKT as a baseline for extending our approach may narrow our comparative analysis. Other current models like AKT, DKVMN could offer different insights or performance metrics. Nonetheless, our choice was informed by DKT's better performance to more recent deep models in related research (Shi et al., 2022; Liu et al., 2022), making it a logical starting point for exploring error predictions.

8 Acknowledgements

This work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Artificial Intelligence under Grant No. 18/CRT/6223 and Grant No. 12/RC/2289-P2 at Insight the Science Foundation Ireland Research Centre for Data Analytics, which is co-funded under the European Regional Development Fund.

References

Ghodai Abdelrahman, Qing Wang, and Bernardo Nunes. 2023. Knowledge tracing: A survey. *ACM Comput. Surv*, 55.

Uri Alon, Meital Zilberstein, Omer Levy, and Eran Yahav. 2019. code2vec: Learning distributed representations of code. *Proceedings of the ACM on Programming Languages*, 3(POPL):1–29.

David Azcona, I-Han Hsiao, and Alan F Smeaton. 2019. Detecting students-at-risk in computer programming classes with learning analytics from students' digital footprints. *User Modeling and User-Adapted Interaction*.

Sahil Bhatia and Rishabh Singh. 2016. Automated correction for syntax errors in programming assignments using recurrent neural networks.

Albert T. Corbett and John R. Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4:253–278.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. 2019. Class-balanced loss based on effective number of samples. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9260–9269.

Andrew Emerson, Fernando J. Rodríguez, Bradford Mott, Andy Smith, Wookhee Min, Kristy Elizabeth Boyer, Cody Smith, Eric Wiebe, and James Lester. 2019. Predicting early and often: Predictive student modeling for block-based programming environments. *International Educational Data Mining Society*.

J Figueiredo and F García-Peñalvo. 2021. Teaching and learning tools for introductory programming in university courses. pages 1–6.

Aritra Ghosh, Neil Heffernan, and Andrew S Lan. 2020. Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2330–2339.

Aritra Ghosh, Jay Raspat, and Andrew Lan. 2021. Option tracing: Beyond correctness analysis in knowledge tracing. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12748 LNAI:137–149.

Sumit Gulwani, Ivan Radiček, and Florian Zuleger. 2018. Automated clustering and program repair for introductory programming assignments. *ACM SIGPLAN Notices*, 53:465–480.

Zhenya Huang, Qi Liu, Chengxiang Zhai, Yu Yin, Enhong Chen, Weibo Gao, and Guoping Hu. 2019. Exploring multi-objective exercise recommendations in online education systems. *International Conference on Information and Knowledge Management, Proceedings*, pages 1261–1270.

Tanja Käser, Severin Klingler, Alexander G Schwing, and Markus Gross. 2017. Dynamic bayesian networks for student modeling. *IEEE Transactions on Learning Technologies*, 10(4):450–462.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2018. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327.

Naiming Liu, Zichao Wang, Richard Baraniuk, and Andrew Lan. 2022. Open-ended knowledge tracing for computer science education.

Qi Liu, Shuanghong Shen, Zhenya Huang, Enhong Chen, Senior Member, and Yonghe Zheng. 2023. A survey of knowledge tracing. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, v3.

Yunfei Liu, Yang Yang, Xianyu Chen, Jian Shen, Haifeng Zhang, and Yong Yu. 2020. Improving knowledge tracing via pre-training question embeddings. *IJCAI International Joint Conference on Artificial Intelligence*, 2021-January:1577–1583.

Rodrigo Pessoa Medeiros, Geber Lisboa Ramalho, and Taciana Pontual Falcao. 2019. A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Transactions on Education*, 62.

Hiromi Nakagawa, Yusuke Iwasawa, and Yutaka Matsuo. 2019. Graph-based knowledge tracing: modeling student proficiency using graph neural network. In *IEEE/WIC/ACM International Conference on Web Intelligence*, pages 156–163.

Sagar Parihar, Ziyaan Dadachanji, Praveen Kumar Singh, Rajdeep Das, Amey Karkare, and Arnab Bhattacharya. 2017. Automatic grading and feedback using program repair for introductory programming courses. *ITiCSE*.

Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas Guibas, Jascha Sohl-Dickstein, Stanford University, and Khan Academy. 2015. Deep knowledge tracing. *Advances in Neural Information Processing Systems*, 28.

Thomas W. Price, David Hovemeyer, Kelly Rivers, Ge Gao, Austin Cory Bart, Ayaan M. Kazerouni, Brett A. Becker, Andrew Petersen, Luke Gusukuma, Stephen H. Edwards, and David Babcock. 2020. Progsnap2: A flexible format for programming process data. *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*, pages 356–362.

Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis Vlahavas. 2011. On the stratification of multi-label data. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part III 22*, pages 145–158. Springer.

Yang Shi, Min Chi, Tiffany Barnes, and Thomas W Price. 2022. Code-dkt: A code-based knowledge tracing model for programming tasks. *Proceedings of the 15th International Conference on Educational Data Mining*.

Dowon Song, Myungho Lee, and Hakjoo Oh. 2019. Automatic and scalable detection of logical errors in functional programming assignments. *Proceedings of the ACM on Programming Languages*, 3.

Ryo Suzuki, Gustavo Soares, Elena Glassman, Andrew Head, Loris D'Antoni, and Björn Hartmann. 2017. Exploring the design space of automatically synthesized hints for introductory programming assignments. *Conference on Human Factors in Computing Systems - Proceedings*, Part F127655:2951–2958.

Michael Thuné and Anna Eckerdal. 2019. Analysis of students' learning of computer programming in a computer laboratory context. *European Journal of Engineering Education*, 44:769–786.

L. Wang, Angela Sy, Larry Liu, and C. Piech. 2017. Learning to represent student knowledge on programming exercises using deep learning. *Educational Data Mining*.

Tong Wu, Qingqiu Huang, Ziwei Liu, Yu Wang, and Dahua Lin. 2020. Distribution-balanced loss for multi-label classification in long-tailed datasets. In *Computer Vision – ECCV 2020*, pages 162–178, Cham. Springer International Publishing.

Yingfei Xiong, Xinyuan Liu, Muhan Zeng, Lu Zhang, and Gang Huang. 2018. Identifying patch correctness in test-based program repair. *Proceedings - International Conference on Software Engineering*, pages 789–799.

Jian Zhang, Xu Wang, Hongyu Zhang, Hailong Sun, Kaixuan Wang, and Xudong Liu. 2019. A novel neural source code representation based on abstract syntax tree. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pages 783–794.

Jiani Zhang, Xingjian Shi, Irwin King, and Dit-Yan Yeung. 2017. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th international conference on World Wide Web*, pages 765–774.

A Appendix

A.1 Error Distribution

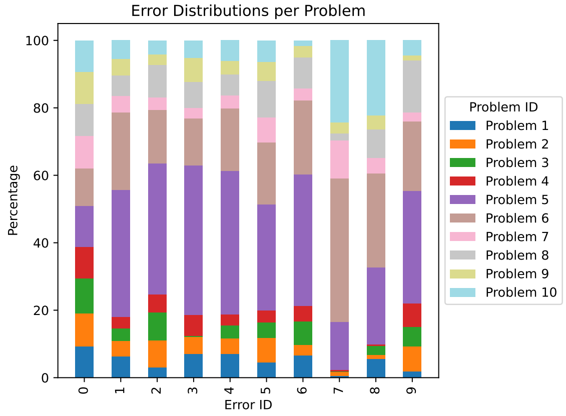


Figure 7: The percentage distribution of each question in the top ten error classes.

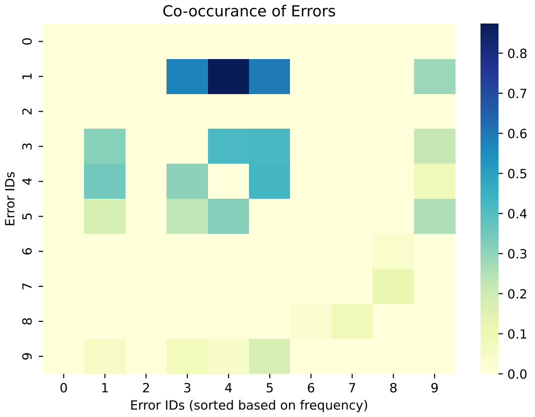


Figure 8: The heatmap showing the co-occurrence of the top ten errors.

Figure 7 maps out the primary distribution of errors across various problems, Figure 8 highlights an intricate aspect of this landscape: the co-occurrence of errors. This heatmap shows how frequently rarer errors appear alongside more common ones, offering insights into error correlations that can influence teaching strategies. Understanding these relationships is key to creating targeted interventions that simultaneously address multiple areas of student difficulty, thus streamlining the path to mastery and enhancing the overall efficacy of programming education.

A.2 Students Attempts

While nearly all students attempted the questions (see Figure 9), there was a notably higher number of attempts on questions five and six, with submissions averaging between 1700 to 2500 as highlighted in Table 5. This increase in attempts corresponded with a higher occurrence of errors in these questions (see Figure 10), suggesting that students were struggling to correct their mistakes, potentially encountering new challenges as they explored different solutions.

A.3 Model Architecture Configuration

Table 6 details the architecture and parameters that define the final models in our study. It encompasses the chosen input features, training configurations, loss functions, and the durations required for both training and inference across each model.

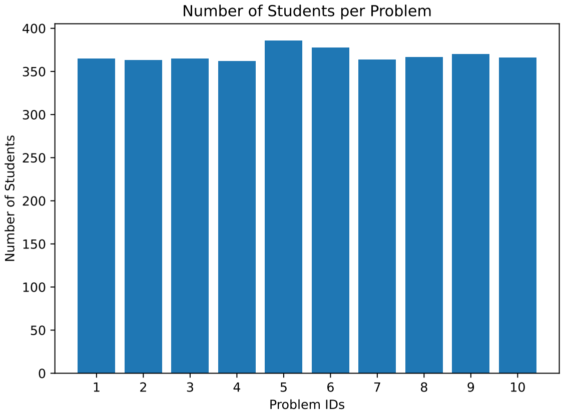


Figure 9: The total numbers of student attempting the 10 questions in assignment one.

Problem ID	Number of Attempts
1	663
2	694
3	699
4	653
5	2578
6	1743
7	678
8	852
9	635
10	800

Table 5: Number of Attempts per Problem

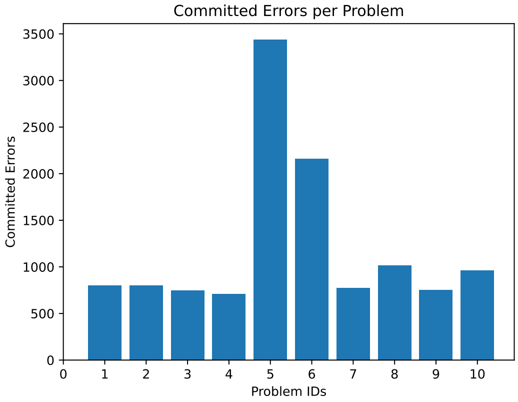


Figure 10: The number of errors committed in each student based on all the attempts.

Model	Input Features	Model Architecture	Output	Training Configurations	Train and inference Time
Simple Baseline	Top errors per problem	-	10	-	1m36s
Complex Baseline	Top errors per problem per attempts	-	10	-	1m36s
HE-BCE	$\mathrm{E_r}(r_t)$, $\mathrm{E_e}(e_t)$, $\mathrm{E_c}(c_t)$	layers=1, hidden=512	10	lr=0.00073, epochs=28, bs=16	7m44s
HE-FL	$\mathrm{E_r}(r_t)$, $\mathrm{E_e}(e_t)$, $\mathrm{E_p}(p_t)$	layers=1, hidden=256	10	lr=0.0004, epochs=70, α(Fl)=0.96, γ(FL)=2.31, bs=16	8m14s
OE-DBloss	$\mathrm{E_r}(r_t)$, $\mathrm{E_e}(e_t)$, $\mathrm{E_{\{ref\}}}(\{ref\}_t)$	layers=1, hidden=256	9	lr=0.000485, epochs=80, α(FL)=0.96, γ(FL)=4.82, β(CB)=0.955, α(DB)=0.93, γ(DB)=0.89, β(DB)=1.53, bs=16	14m14s
OE-BCE	$\mathrm{E_r}(r_t)$, $\mathrm{E_e}(e_t)$, $\mathrm{E_{\{ref\}}}(\{ref\}_t)$	layers=1, hidden=256	9	lr=0.0009, epochs=80, bs=16	10m44s
IE-BCE	$\mathrm{E_r}(r_t)$, $\mathrm{E_e}(e_t)$, $\mathrm{E_{\{ref\}}}(\{ref\}_t)$	layers=1, hidden=256	9	lr=0.0009, epochs=80, bs=16	25m32s
DKT	$\mathrm{E_r}(r_t)$	layers=1, hidden=512	1	lr=0.0005, epochs=50, bs=16	8m35s
OE-BCE*	$\mathrm{E_r}(r_t)$, $\mathrm{E_e}(e_t)$, $\mathrm{E_p}(p_t)$	layers=2, hidden=256	9	lr=0.00071, epochs=70, bs=16	6m32s

Table 6: Model architecture configurations for various prediction task and the set of best input features.

Improving Readability Assessment with Ordinal Log-Loss

Ho Hung Lim and **John S. Y. Lee**
Department of Linguistics and Translation
City University of Hong Kong
limhhresearch@gmail.com,jsylee@cityu.edu.hk

Abstract

Automatic Readability Assessment (ARA) aims to predict the level of difficulty of a text, e.g. at Grade 1 to Grade 12. It can be helpful for teachers and students in identifying and revising text to the desirable level of difficulty. ARA is an ordinal classification task since the predicted levels follow an underlying order, from easy to difficult. However, most neural ARA models ignore the distance between the gold level and predicted level, treating all levels as independent labels. This paper investigates whether distance-sensitive loss functions can improve ARA performance. We evaluate a variety of loss functions on neural ARA models, and show that ordinal log-loss can produce statistically significant improvement over the standard cross-entropy loss in terms of adjacent accuracy in a majority of our datasets.

1 Introduction

Automatic Readability Assessment (ARA) aims to predict the level of difficulty of a text, e.g. at Grade 1 to Grade 12. It can be helpful for teachers and students in identifying and revising text to the desirable level of difficulty. ARA is an ordinal classification task since the levels follow an underlying order, from easy to difficult. Yet, in ARA models trained with traditional machine learning, the use of ordinal classification has yielded mixed results (Heilman et al., 2008; Feng et al., 2010; Jiang et al., 2014). Further, most neural ARA models treat the task as multi-class classification (Xia et al., 2016; Azpiazu and Pera, 2019; Filighera et al., 2019; Tseng et al., 2019; Deutsch et al., 2020; Martinc et al., 2021; Lee et al., 2021) and ignore the distance between the gold level and predicted level. In these models, a classifier is typically trained with the standard cross-entropy loss function, which treats the difficulty levels as independent labels. Further, performance evaluation often penalizes incorrect predictions equally, regardless of their distance from the gold.

Recognizing the ordinal nature of ARA could potentially enhance performance and enable more accurate evaluation. A loss function that reflects label distance could be suitable, since the boundary between difficulty levels may not be clear-cut, especially on fine-grained scales. While severe mistakes are never desirable, a sufficiently close prediction may be acceptable in some applications, such as retrieval of extra-curricular reading materials. Evaluation metrics that reflect the average distance from the gold label would therefore be more informative.

Distance-sensitive loss functions have received relatively little attention in neural ARA. Zeng et al. (2022) showed that soft labels could improve performance, but the evaluation was limited to BERT and only one loss function. We present a more comprehensive study on a variety of loss functions, evaluated on a range of pre-trained language models, hyper-parameters, and performance metrics. Experimental results show that ordinal log-loss (Castagnos et al., 2022) performs best overall for neural ARA models. It achieves a statistically significant improvement over the standard cross-entropy loss in terms of adjacent accuracy in a majority of our datasets, though sometimes at the expense of accuracy.

The rest of the paper is organized as follows. After a review of the major loss functions in Section 2, we give details on the experimental set-up in Section 3. We then report results in Section 4.[1]

2 Previous work

Many text classification tasks, ranging from ARA and essay scoring, to sentiment and review rating prediction, have an ordinal structure. Let $\mathcal{Y} = \{r_1, r_2, ..., r_K\}$ be the set of possible labels. Ordinal binary classification exploits the structure

[1] Code and data can be accessed at https://github.com/hhlim333/Readability-Assessment-with-Ordinal-Log-Loss

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 343–350
June 20, 2024 ©2024 Association for Computational Linguistics

with $K-1$ binary classifiers (Frank and Hall, 2001). Ordinal Multi-class Classification with Voting was found to be potentially helpful in improving ARA performance (Jiang et al., 2014). Ordinal regression models have been applied to ARA models trained in traditional machine learning. While Heilman et al. (2008) found that the Proportional Odds Model offered competitive performance, Feng et al. (2010) reported that ordinal classifiers did not perform better than standard classifiers. Loss-sensitive classification, which is the focus of this paper, utilizes loss functions that impose higher penalty to predictions further from the gold label, based on a distance function $d(r_i, r_j)$ that specifies the distance between labels r_i and r_j. Two main families of these loss functions are as follows.

2.1 Soft labels

Soft labels for ordinal regression (Bertinetto et al., 2020) is a distance-sensitive loss function that has been found to be effective for ARA. The soft label is defined as follows:

$$y_i = \frac{\exp\left(-\beta \cdot d(r_i, r_t)\right)}{\sum_{k=1}^{K} \exp\left(-\beta \cdot d(r_k, r_t)\right)} \tag{1}$$

where r_t is the gold label; $r_i \in \mathcal{Y}$ is the i-th label; and the hyperparameter β specifies how much more probability mass to assign to labels closer to the gold.

Zeng et al. (2022) applied the soft label version of Diaz and Marathe (2019) to ARA using a simple distance function: the distance between the gold and an adjacent label is a positive constant, and infinity for all other labels. A BERT-based neural classifier trained on this loss function outperformed the standard cross-entropy loss on both English and Chinese data.

2.2 Ordinal log-loss

Ordinal log-loss (OLL) is defined as:

$$-\sum_{i=1}^{N} \log(1-p_i) d(y,i)^{\alpha} \tag{2}$$

where the hyperparameter α adjusts the amount of penalty, with a higher value leading to the greater penalty for predicted labels at a longer distance from the gold (Castagnos et al., 2022). OLL is distinguished in its use of the weight $-log(1-p_i)$, rather than p_i as in many other loss functions, to impose greater penalty on more severe errors.

Castagnos et al. (2022) have shown OLL to be beneficial in a number of text classification tasks, but their evaluation focused only on BERT-tiny. This paper is the first attempt to apply OLL on ARA. We conduct a comprehensive study utilizing a variety of loss functions and pre-trained language models, and analyzing trade-off between accuracy and adjacent accuracy.

3 Experimental Set-up

This section describes the loss functions (Section 3.1), the datasets (Section 3.2) and training procedure (Section 3.3).

3.1 Loss functions

We investigate the following loss functions for training neural ARA models:[2]

Baseline The standard cross-entropy loss.

WKL Weighted Kappa Loss (de la Torre et al., 2018).

EMD Earth Mover's Distance (Hou et al., 2016).

OLL-α Ordinal log-loss (Castagnos et al., 2022) with the hyperparameter α, as defined in Section 2.2.

SOFT-β Soft labels (Bertinetto et al., 2020) with the hyperparameter β, as defined in Section 2.1.

Zeng et al The model proposed by Zeng et al. (2022) (Section 2.1), based on the soft label version of Diaz and Marathe (2019), which does not use the β hyperparameter.

Following Castagnos et al. (2022), we tuned the α parameter for OLL on $\{1, 1.5, 2\}$ and the β parameter for SOFT on $\{2, 3, 4\}$. They were optimized on the validation set of the Cambridge Dataset to $\alpha = 1$ and $\beta = 2$, respectively. We used the default distance function $d(r_i, r_j) = |r_i - r_j|$ in all experiments.

3.2 Datasets

Our experiments make use of three English and two Chinese datasets (see detailed statistics in Appendix A):

[2] https://github.com/glanceable-io/ordinal-log-loss

Loss function	Cam		CC		OSE		CMT		CMER	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
Baseline	0.387	0.533	1.047	1.729	**0.042**	0.077	1.244	3.524	1.696	5.666
Zeng et al	0.347	0.413	0.953	1.494	0.056	0.084	1.118	2.985	1.681	5.623
OLL-1	0.347	**0.400**	**0.776**	**1.012**	0.074	0.13	**1.112**	**2.894**	**1.638**	**4.847**
SOFT-2	**0.333**	**0.400**	1.035	1.694	**0.042**	0.077	1.159	3.169	1.679	5.592
EMD	0.433	0.553	0.906	1.541	0.046	**0.06**	1.171	3.104	1.664	5.205
WKL	0.867	1.493	1.235	2.671	0.446	0.614	2.252	10.107	3.455	19.177

Table 1: Mean Absolute Error (MAE) and Mean Squared Error (MSE) in ARA using RoBERTa on the English datasets Cambridge (Cam), Common Core (CC) and OneStopEnglish (OSE); and using MacBERT on the Chinese datasets CMT and CMER

Cambridge (Cam) This dataset contains articles for various Cambridge English Exams, labeled with five levels (A2-C2) in the Common European Framework of Reference (Xia et al., 2016). We use the train/validation/test set of the downsampled version provided by Lee et al. (2021), which consists of 60 items per level.[3]

OneStopEnglish (OSE) This corpus consists of 189 aligned texts, each written at three reading levels: beginner, intermediate, and advanced (Vajjala and Lučić, 2018), hence a total of 567 texts.[4]

Common Core (CC) The Common Core corpus consists of 168 texts, labeled at five grade bands (Grades 2–3, 4–5, 6–8, 9-10, and 11–12) from Appendix B of the English Language Arts Standards of the Common Core State Standards (Chen and Meurers, 2016).[5]

China Mainland Textbook (CMT) This corpus consists of a total of 2,723,430 characters, distributed in 2,621 texts in twelve grades, all taken from Chinese textbooks from the first grade of primary school to the third grade of high school in mainland China (Cheng et al., 2019).

China Mainland Extracurricular Reading (CMER) This corpus consists of 2,260 texts distributed at Grade 1 to 12, taken from extracurricular reading books for children and teenagers.[6]

3.3 Pre-trained language models

We evaluated the pre-trained language models BERT, RoBERTa, BART, and XLNET[7] in English experiments. In the Chinese experiments, we used MacBERT[8], which was shown to perform best in previous research on Chinese ARA (Lim et al., 2022). All models were downloaded from HuggingFace transformers v4.5.0 (Wolf et al., 2020).[9]

4 Experimental results

All results are averaged based on stratified 5-fold cross-validation with a 8:1:1 split for train/validation/test. We first report overall results based on Mean Absolute Error (MAE) and Mean Squared Error (MSE) (Section 4.1), and then analyze their performance in terms of adjacent accuracy and accuracy.[10] Henceforth, all Chinese results are based on MacBERT, and the English results on RoBERTa, sicne they performed best among the four PLMs evaluated (see Table 7 in Appendix D).

4.1 Mean Error

Table 1 shows the performance of neural ARA models in terms of MAE and MSE when trained with the loss functions described in Section 3.1. Weighted Kappa Loss (WKL) produced the worst performance, below the standard cross-entropy baseline in all datasets. Earth Mover's Distance (EMD) outperformed the baseline in four out of

[3] Accessed at https://github.com/brucewlee/

[4] Accessed at https://github.com/nishkalavallabhi/

[5] https://xiaobin.ch/Chen_Meurers_16Frequency/

[6] https://github.com/JinshanZeng/DTRA-Readability

[7] https://huggingface.co/bert-base-uncased,roberta-base,bart-base,xlnet-base-cased

[8] https://huggingface.co/hfl/chinese-macbert-large

[9] We used AdamW (optimizer) (Kingma and Ba, 2015), linear (scheduler), 10% (warmup steps), 8 (batch size), 3 (epoch) for all pre-trained language models. For English experiments, we use the learning rate of 2e-5 for BERT and 3e-5 for the other pre-trained language models. For Chinese experiments,we use the learning rate of 2e-5 for MacBERT.

[10] All metrics are calculated with SciKit-learn (Pedregosa et al., 2011).

Loss	(a) Accuracy					(b) Adjacent Accuracy				
Function	Cam	CC	OSE	CMT	CMER	Cam	CC	OSE	CMT	CMER
Baseline	0.68	0.294	0.975	0.364	0.285	0.940	0.659	0.982	0.686	0.561
Zeng et al	0.68	0.318	0.958	0.382	0.277	0.98	0.729	0.986	0.735	0.575
OLL-1	0.673	0.341	0.954	0.368	0.232	**0.987***	**0.882****	0.972	**0.740***	0.563
OLL-1.5	0.64	0.329	0.846	0.316	0.209	0.973	**0.882****	**0.993**	0.738*	0.573
OLL-2	0.56	0.341	0.891	0.317	0.201	0.98	0.824**	0.989	0.731*	**0.583**
SOFT-2	0.693	0.294	0.975	0.381	0.277	0.98	0.671	0.982	0.718*	0.574
SOFT-3	**0.727**	0.294	**0.979**	**0.387**	0.281	0.967	0.682	0.986	0.726*	0.555
SOFT-4	0.713	0.294	**0.979**	0.367	**0.29**	0.96	0.659	0.982	0.699	0.568
EMD	0.62	**0.376**	0.961	0.359	0.243	0.953	0.753	0.993	0.709	0.573
WKL	0.387	0.271	0.639	0.182	0.105	0.8	0.659	0.916	0.5	0.307

Table 2: ARA performance based on (a) accuracy; and (b) adjacent accuracy (* means a statistically significant improvement at $p < 0.05$ according to McNemar's Test over the baseline; ** means statistically significant improvement over both the baseline and the Zeng et al. model)

five datasets, yielding the lowest MSE on OSE. The Zeng et al model improved upon the baseline in all datasets except OSE. SOFT-2 outperformed Zeng et al in three out of the four datasets, and produced the best performance on Cambridge (tied with OLL-1), suggesting the utility of the β hyperparameter. Overall, OLL-1 achieved the best performance, with the smallest MSE on four of the five datasets. In the remainder of the discussion, we will focus on Zeng et al, SOFT-β and OLL-α.

4.2 Adjacent accuracy

Table 2(b) shows the results in terms of adjacent accuracy. The OLL-α models outperformed the baseline in the vast majority of settings, suggesting their ability to reduce severe ARA errors.[11] Of the four PLMs, the best performance was obtained with RoBERTa (Appendix D).

OLL-1 achieved the best adjacent accuracy at 0.987 on Cambridge and 0.882 on Common Core.[12] It also scored the highest Macro F1 and Weighed F1 on these two datasets (see Table 5 in Appendix C). OSE is particularly challenging since the baseline already achieved excellent performance at 0.989 adjacent accuracy; OLL was able to make an improvement on adjacent accuracy and F1 only when α is set to 1.5. OLL-1 improved upon the baseline on both Chinese datasets, and outperformed Zeng et al on CMT.

4.3 Accuracy

OLL-1 generally performed worse than the baseline, both in terms of accuracy (Table 2(a))[13] and F1 (Table 6 in Appendix C). SOFT-2 improved upon the baseline and Zeng et al in most settings, although the improvement was not statistically significant.

SOFT-3 established a new state-of-the-art in accuracy and F1 for neural ARA models, on both the Cambridge and OSE datasets. Its performance (accuracy at 0.727 and 0.979, respectively) surpassed the previous best (0.680 and 0.975) in neural models (Lee et al., 2021), although it is still outperformed by hybrid models, which require handcrafted linguistic features. SOFT-3 also obtained the best result in Chinese on CMT (0.387), outperforming the baseline and the Zeng et al model.

5 Conclusion

Since ARA is an ordinal classification task, the magnitude of classification error should in principle be taken into account. This paper has presented a comprehensive evaluation of a variety of loss functions that are sensitive to the distance between the predicted label and gold label.

Our experiments on neural ARA models suggest that ordinal log-loss (OLL) is able to capture the ordinal nature of the task, reducing the mean absolute error and mean squared error on most datasets. It produces significant improvement over the standard cross-entropy function in terms of adjacent accuracy, but at the expense of accuracy in some

[11] Among all combinations of α values, PLMs and datasets, there are only two exceptions: OLL-1 with RoBERTa on OSE, and with BART on Cambridge.

[12] Statistically significant at $p = 0.0391$ and $p = 0.0000$, respectively, according to McNemar's Test.

[13] We obtained slightly higher accuracy for the baseline on the OSE dataset than reported by Lee et al. (2021).

settings. These results suggest that future ARA models should consider using OLL for applications that need to avoid severe errors but do not require precise classification.

A number of research directions may be pursued. First, ARA accuracy could be further improved by optimizing the distance function in the ordinal log-loss and soft label models. Second, the usability of the ARA model in an educational setting, for example assisting teachers and students in text selection and revision, is also worth investigating.

Acknowledgements

This work was partly supported by the Language Fund from the Standing Committee on Language Education and Research (project EDB(LE)/P&R/EL/203/14) and by a Teaching Development Grant from City University of Hong Kong (project 6000834).

References

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.

Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas A. Lord. 2020. Making better mistakes: Leveraging class hierarchies with deep networks. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

François Castagnos, Martin Mihelich, and Charles Dognin. 2022. A Simple Log-based Loss Function for Ordinal Text Classification. In *Proc. 29th International Conference on Computational Linguistics (COLING)*, pages 4604–4609.

Xiaobin Chen and Detmar Meurers. 2016. Characterizing Text Difficulty with Word Frequencies. In *Proc. 11th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*, page 84–94.

Yong Cheng, Dekuan Xu, and Xueqiang Lv. 2019. Automatically Grading Text Difficulty with Multiple Features. *Data Analysis and Knowledge Discovery*, 3(7):103–112.

Jordi de la Torre, Domenec Puig, and Aida Valls. 2018. Weighted kappa loss function for multi-class classification of ordinal data in deep learning. *Pattern Recognition Letters*, 105:144–154.

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*.

Raul Diaz and Amit Marathe. 2019. Soft labels for ordinal regression. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, page 4738–4747.

Lijun Feng, Martin Jansche, Matt Huenerfauth, and Noemie Elhadad. 2010. A Comparison of Features for Automatic Readability Assessment. In *Proc. COLING*.

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.

Eibe Frank and Mark Hall. 2001. A simple approach to ordinal classification. In *Proc. 12th European Conference on Machine Learning (ECML)*, page 145–156.

Michael Heilman, Kevyn Collins-Thompson, and Maxine Eskenazi. 2008. An Analysis of Statistical Models and Features for Reading Difficulty Prediction. In *Proc. Third Workshop on Innovative Use of NLP for Building Educational Applications*.

Le Hou, Chen-Ping Yu, and Dimitris Samaras. 2016. Squared earth mover's distance-based loss for training deep neural networks. In *arXiv preprint arXiv:1611.05916*.

Zhiwei Jiang, Gang Sun, Qing Gu, and Daoxu Chen. 2014. An Ordinal Multi-class Classification Method for Readability Assessment of Chinese Documents. *LNAI*, 8793:61–72.

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*.

Ho Hung Lim, Tianyuan Cai, John S. Y. Lee, and Meichun Liu. 2022. Robustness of Hybrid Models in Cross-domain Readability Assessment. In *Proc. 20th Workshop of the Australasian Language Technology Association (ALTA)*.

Matej Martinc, Senja Pollak, Marko, and Robnik-Šikonja. 2021. Supervised and Unsupervised Neural Approaches to Text Readability. *Computational Linguistics*, 47(1):141–179.

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.

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*.

	Cam		CC		OSE	
Grade	Texts	Text length	Texts	Text length	Texts	Text length
1	60	140.12	20	294.65	189	531.97
2	60	271.25	30	320.70	189	677.90
3	60	614.50	45	472.09	189	820.76
4	60	778.73	36	549.83	na	na
5	60	761.85	37	612.05	na	na

Table 3: Number of texts and average length at each grade in the Cam, CC and OSE dataset

	CMT		CMER	
Grade	Texts	Text length	Texts	Text length
1	235	108.95	218	145.53
2	320	198.58	217	308.44
3	386	329.48	234	538.35
4	321	425.39	229	628.08
5	282	569.82	200	682.41
6	252	660.89	255	701.29
7	199	1202.13	221	1227.19
8	142	1176.94	205	1324.25
9	134	1443.84	188	1302.54
10	140	1617.08	100	2182.08
11	89	1900.85	96	2252.34
12	121	1930.74	97	2043.69

Table 4: Number of texts and average length at each grade in the CMT and CMER dataset

Sowmya Vajjala and Ivana Lučić. 2018. OneStopEnglish corpus: A new corpus for automatic readability assessment and text simplification. In *Proc. 13th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 297–304.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. In *Proc. Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, page 38–45.

Menglin Xia, Ekaterina Kochmar, and Ted Briscoe. 2016. Text readability assessment for second language learners. In *Proc. 11th Workshop on Innovative Use of NLP for Building Educational Applications*, page 12–22.

Jinshan Zeng, Yudong Xie, Xianglong Yu, John S. Y. Lee, and Ding-Xuan Zhou. 2022. Enhancing Automatic Readability Assessment with Pre-training and Soft Labels for Ordinal Regression. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4586–4597.

A Appendix: Dataset statistics

This section provides detailed statistics for all datasets.

B Appendix: Computing details

We used a NVIDIA Tesla V100 GPU to train 80% of the full dataset. The following is the total training time of the experiments on OLL-1, measured in seconds:

English Experiments (BERT, RoBERTa, XLNet, BART):

- Cambridge (638,496,1410,607)
- OneStopEnglish (1261, 948, 2286, 1110)
- CommonCore (382,310,776,377)

Chinese Experiment (MacBERT):

- CMT (12498)
- CMER (11809)

C Appendix: F1 Evaluation

This section reports F1 evaluation, based on adjacent accuracy (Table 5) and accuracy (Table 6), respectively. We used RoBERTa on the English datasets Cambridge (Cam), Common Core (CC) and OneStopEnglish (OSE); and MacBERT on the Chinese datasets CMT and CMER.

D Appendix: Evaluation on other PLMs

This appendix provides detailed results for all pretrained language models (BERT, RoBERTa, XLNet, BART).

Loss	Macro F1					Weighted F1				
Function	Cam	CC	OSE	CMT	CMER	Cam	CC	OSE	CMT	CMER
Baseline	0.938	0.527	0.982	0.593	0.518	0.938	0.551	0.982	0.655	0.548
Zeng et al	0.98	0.615	0.986	0.647	0.532	0.98	0.658	0.986	0.715	0.562
OLL-1	**0.987**	**0.839**	0.972	0.642	0.502	**0.987**	**0.862**	0.972	0.721	0.544
OLL-1.5	0.973	0.833	**0.993**	**0.661**	0.513	0.973	0.859	**0.993**	**0.722**	0.554
OLL-2	0.98	0.742	0.989	0.631	0.518	0.98	0.788	0.989	0.712	0.563
SOFT-2	0.98	0.544	0.982	0.629	**0.533**	0.98	0.572	0.982	0.694	0.562
SOFT-3	0.966	0.557	0.986	0.636	0.511	0.966	0.588	0.986	0.705	0.539
SOFT-4	0.959	0.527	0.982	0.605	0.528	0.959	0.551	0.982	0.673	0.555
EMD	0.952	0.722	**0.993**	0.62	0.53	0.952	0.722	**0.993**	0.686	**0.565**
WKL	0.766	0.626	0.894	0.437	0.216	0.766	0.605	0.894	0.473	0.248

Table 5: ARA performance in F1, based on *adjacent accuracy*

Loss	Macro F1					Weighted F1				
Function	Cam	CC	OSE	CMT	CMER	Cam	CC	OSE	CMT	CMER
Baseline	0.658	0.091	0.975	0.282	0.253	0.658	0.134	0.975	0.324	0.27
Zeng et al	0.668	0.131	0.958	**0.322**	0.246	0.668	0.173	0.958	**0.363**	0.262
OLL-1	0.654	0.206	0.954	0.279	0.201	0.654	0.242	0.954	0.346	0.221
OLL-1.5	0.591	0.189	0.812	0.236	0.167	0.591	0.226	0.812	0.286	0.181
OLL-2	0.496	0.182	0.868	0.227	0.153	0.496	0.224	0.868	0.273	0.168
SOFT-2	0.68	0.095	0.975	0.305	0.246	0.68	0.139	0.975	0.351	0.262
SOFT-3	**0.717**	0.093	**0.979**	0.318	0.253	**0.717**	0.136	**0.979**	0.361	0.264
SOFT-4	0.699	0.091	**0.979**	0.294	**0.259**	0.699	0.134	**0.979**	0.337	**0.274**
EMD	0.569	**0.243**	0.961	0.288	0.204	0.569	**0.284**	0.961	0.329	0.214
WKL	0.237	0.157	0.539	0.083	0.024	0.237	0.139	0.539	0.08	0.026

Table 6: ARA performance in F1, based on *accuracy*

Metric →		Accuracy			Adjacent Accuracy		
PLM	Loss Func.	Cam	CC	OSE	Cam	CC	OSE
BERT	Baseline	0.573	0.388	**0.919**	0.907	0.694	0.989
	Zeng et al	0.567	**0.4**	0.719	0.94	**0.835**	0.993
	OLL-1	0.5	0.365	0.709	**0.973***	0.812*	0.989
	OLL-1.5	0.44	0.365	0.737	**0.973***	0.788*	**0.996**
	OLL-2	0.467	0.353	0.705	**0.973***	0.765	0.993
	SOFT-2	**0.593**	0.388	0.768	0.913	0.753	0.993
	SOFT-3	0.573	0.376	0.765	0.913	0.718	0.993
	SOFT-4	0.587	0.353	0.754	0.92	0.659	0.993
	WKL	0.407	0.318	0.505	0.813	0.753	0.863
	EMD	0.48	0.353	0.786	0.92	0.776	0.993
RoBERTa	Baseline	0.68	0.294	0.975	0.94	0.659	0.982
	Zeng et al	0.68	0.318	0.958	0.98	0.729	0.986
	OLL-1	0.673	**0.341**	0.954	**0.987***	**0.882****	0.972
	OLL-1.5	0.64	0.329	0.846	0.973	**0.882****	**0.993**
	OLL-2	0.56	**0.341**	0.891	0.98	0.824**	0.989
	SOFT-2	0.693	0.294	0.975	0.98	0.671	0.982
	SOFT-3	**0.727**	0.294	**0.979**	0.967	0.682	0.986
	SOFT-4	0.713	0.294	**0.979**	0.96	0.659	0.982
	WKL	0.387	0.271	0.639	0.8	0.659	0.916
	EMD	0.62	0.376	0.961	0.953	0.753	0.993
BART	Baseline	0.62	0.388	**0.968**	0.927	0.776	0.989
	Zeng et al	0.593	**0.435**	0.944	0.92	0.788	**0.996**
	OLL-1	0.52	0.353	0.965	0.92	0.847	0.993
	OLL-1.5	0.493	0.318	0.958	**0.94**	0.871**	0.993
	OLL-2	0.42	0.294	0.916	**0.94**	**0.882****	**0.996**
	SOFT-2	0.6	0.412	0.947	0.92	0.776	0.993
	SOFT-3	0.6	**0.435**	0.944	0.9	0.8	0.986
	SOFT-4	**0.627**	0.388	0.954	0.907	0.776	0.989
	WKL	0.393	0.294	0.596	0.8	0.612	0.902
	EMD	0.56	0.4	0.961	0.913	0.788	0.993
XLNET	Baseline	0.573	0.365	0.804	0.933	0.671	0.993
	Zeng et al	**0.713**	0.388	0.811	0.933	0.8	0.993
	OLL-1	0.653	0.318	0.737	0.967	0.824*	**0.996**
	OLL-1.5	0.593	0.365	**0.818**	**0.973****	**0.847***	0.993
	OLL-2	0.467	0.329	0.807	**0.973****	0.835*	0.993
	SOFT-2	0.667	0.388	0.877	0.933	0.753	0.993
	SOFT-3	0.653	**0.424**	0.891	0.92	0.741	0.996
	SOFT-4	0.633	0.341	0.853	0.933	0.753	0.993
	WKL	0.42	0.318	0.481	0.86	0.659	0.86
	EMD	0.587	0.318	0.856	0.9	0.741	0.989

Table 7: ARA performance on the English datasets (* means statistically significant improvement at $p < 0.05$ according to McNemar's Test over the baseline; ** means statistically significant improvement over both baseline and Zeng et al.)

Automated Sentence Generation for a Spaced Repetition Software

Benjamin Paddags **Daniel Hershcovich** **Valkyrie Savage**
Department of Computer Science, University of Copenhagen
{bepa, dh, vasa}@di.ku.dk

Abstract

This paper presents and tests AllAI, an app that utilizes state-of-the-art NLP technology to assist second language acquisition through a novel method of sentence-based spaced repetition. Diverging from current single word or fixed sentence repetition, AllAI dynamically combines words due for repetition into sentences, enabling learning words in context while scheduling them independently. This research explores various suitable NLP paradigms and finds a few-shot prompting approach and retrieval of existing sentences from a corpus to yield the best correctness and scheduling accuracy. Subsequently, it evaluates these methods on 26 learners of Danish, finding a four-fold increase in the speed at which new words are learned, compared to conventional spaced repetition. Users of the retrieval method also reported significantly higher enjoyment, hinting at a higher user engagement.

1 Introduction

Spaced repetition is a well-known learning technique that involves repeated exposure to learning material, usually at increasing intervals, which has been shown to enhance long-term retention (see section 2.1). Usually, spaced repetition in language learning is done by repeating single words or whole sentences curated by humans. Already a decade ago, the potential of computational linguistics for vocabulary learning was identified by Zock et al. (2014, p. iii): "There is so much more we could do these days by using corpora and computational linguistics know-how, to extract the to-be learned words from text and to display them with their context. Hence, rather than having the user repeat single words (or word pairs) we could display them in various contexts (e.g. sentences), thereby making sure that the chosen ones correspond to the learners' level and interests.." Developing a software system that automatically generates sentences for spaced repetition has the potential to provide learners with a more efficient learning experience by generating sentences with many words that are due for repetition, with more personalized and versatile tasks that make studying more enjoyable and engaging. Furthermore, it could free up human language teachers to focus on in-person teaching instead of writing example sentences.

This work introduces AllAI (Automated Language Learning with AI), an application utilizing NLP to create such a sentence-based approach to spaced repetition. The app keeps track of the user's vocabulary and generates sensible sentences (spaced repetition "tasks") from only the subset of words of a language that the user knows and currently needs to repeat, with some minor amount of new words that make sense to learn. The user can then calibrate the spaced repetition of each word by answering which of the words in the sentence they correctly remembered. We then investigate the learning outcomes of using such a system compared to current solutions. As such, the main research questions are the following:

1. Which NLP paradigm and configuration can optimize spaced repetition timing and best avoid out-of-user-vocabulary words, while retaining high correctness of the generated sentences?

2. How does sentence-based spaced repetition using the best-performing options from the first question influence user engagement and learning outcomes among language learners, compared to conventional approaches?

The proposed system combines the following potential advantages over the conventional spaced repetition approaches mentioned in 2:

1. It honors the minimum information principle.

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 351–364
June 20, 2024 ©2024 Association for Computational Linguistics

2. It shows words in context for a less artificial learning situation and the possibility to infer meaning.
3. It can generate a variety of tasks for high novelty value.
4. It could be optimized for additional objectives, such as entertainment value (e.g. subsequent sentences could form a story), variety of grammar, or others.

The main contribution of this work is putting the current and soon-to-be due words of a spaced repetition system into context by investigating different methods of automating the forming of sentences with them. We also develop a metric for calculating the scheduling accuracy and select other metrics to assess the quality of the output sentences for the task. We compare a range of candidate methods and configurations that managed to return sensible sentences containing target words with regard to these metrics. We develop an application consisting of a front-end for the user to interact with the generated tasks and a back-end to do the spaced repetition scheduling and house the developed methods for sentence generation. Finally, we test the real-world usefulness of two of the best-performing methods, a retrieval-based method and a GPT-3.5-based method using few-shot prompting, in a user study, assessing learning outcomes and indicators of user engagement against a baseline similar to current spaced repetition practices.

We implement and test the system in Danish. Still, it applies to any language in which the sentences are made up of words and is developed in such a way that it could teach a different language if the NLP component is swapped out, e.g. by translating the prompts of a prompting-based solution to a different language.

2 Background and Related Work

2.1 Spaced repetition

Previous research has found a large beneficial effect of computer-assisted language learning (CALL) on vocabulary learning (Hao et al., 2021). One possible CALL technique is spaced repetition. Spaced repetition means reviewing information that one wants to remember repeatedly and with temporal spacing between each exposure to the same information. A review usually involves the learner being prompted, trying to recall, and then getting feedback. It has been shown to produce better learning than immediate repetition without spacing, e.g. in this meta-analysis by Carpenter et al. (2012) for spacing in general. Based on the idea of physical flashcards with a prompt on one side and the correct answer on the other, that are reviewed at increasing intervals (Leitner, 1972), most spaced repetition software (e.g. Anki (Elmes), shown as an example in figure 1, Mnemosyne (Çakmak et al., 2021), SuperMemo (Wozniak)) usually show a memory recall task to the user and expect the user to try to solve it. Thereafter, the solution is shown, and the user rates how well they could recall it. The system uses the recall quality to calculate the spacing until the task is presented to the user again, which should ideally be right before the user is likely to forget it.

In the context of language learning, spaced repetition can be used for the parts of L2 acquisition that require memorization, such as vocabulary learning. There are thus three common approaches for vocabulary retention using spaced repetition systems, as evidenced by the kinds of card decks users have published for the Anki app [1]. The first one is to use single pieces of vocabulary as the task, the second one is to use whole sentences or text snippets, and the third one is to use single words, but with one or more example sentences also provided on either the solution side or both sides of the flashcard. The main argument for the first practice is the minimum information principle: Each task should be as minimal as possible, ideally one piece of information (Jankowski, 1999), allowing for independent scheduling of each of the bits of knowledge. On the other hand, language is naturally used in context, where words learned in the context of a sentence reinforce each other, strengthening thus learning and recall, meaning that remembering words out of context is a very artificial task and much harder than if related words are present which can give hints about the meaning (Ramos and Dario, 2015). This work sets itself apart from the existing literature on spaced repetition by examining the effects of integrating a sentence generation component that generates sentences for single use on demand, which makes it possible to keep scheduling single words and adhering to the minimum information principle while showing words in context.

[1] "Shared Decks" https://ankiweb.net/shared/decks/danish

2.2 Language models, Text Generation and Language Teaching

A central concept in NLP is the language model (LM): A statistical model that assigns a probability to any possible sequence of tokens (Jurafsky and Martin, 2023). This probability distribution can be sampled, thereby generating text. The ability of language models to generate fluent text has significantly advanced in recent years, to the point where they can create text of human-like quality (Fatima et al., 2022).

With the strong performance of transformer-based pre-trained models (PLMs), such as GPT-3 (Brown et al., 2020), zero- or few-shot prompting of these PLMs have gained popularity, profiting from the excellent general understanding of the semantics and syntax of language that they can develop through pre-training on large and diverse text corpora. Recent research has demonstrated that especially for very large LMs, prompting approaches can reach similar results to fine-tuning-based approaches on many NLP tasks, or even outperform them (Wei et al., 2022; Brown et al., 2020).

Even before the advent of modern language models, Brown et al. (2005) used a corpus of words with example sentences to generate cloze questions with a keyword missing, which the user has to fill in, to assess language learners' level. This is similar to the task this work tries to achieve: generating sentences based on multiple words that should be contained. However, they only use one input word which in their database is already associated with sample sentences, so the exact approach cannot be copied for multiple input words. However, using a retrieval system on a corpus of example sentences can be a viable approach since queries can consist of multiple words. When it comes to using LMs in second language teaching, Okano et al. (2023) try a reinforcement learning approach, as well as a few-shot prompting approach to make large language models output sentences containing specific grammatical structures and find that both approaches are feasible. Their research was published after this paper's experiments were finished, so it could not be used for inspiration. While they focus on generating sentences with specific grammatical structures, this work instead tries to achieve the use of specific words in the sentence, which is easier in the sense that instead of transferring implicit grammatical patterns, the model just needs to use the same words already given in the input, but harder in the sense that there are thousands of words that might need to be generated, while Okano et al. (2023) only had 20 grammatical structures to optimize for. There have also been successful attempts at creating flashcards for spaced repetition systems using LLMs, such as Gossmann (2024), Cruz (2023) and Velde (2023). Gossmann and Cruz focus on summarizing knowledge from articles into flashcards while Velde is applying their approach to vocabulary learning. Differently from what we are attempting, their flashcards are static, so they will still always show each word in context of the same information, which is equivalent to the third existing approach mentioned in section 2.1.

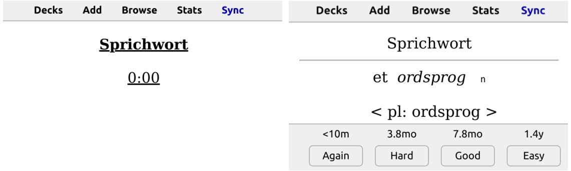


Figure 1: The Anki spaced repetition system, step by step: a task is presented (left), the solution (translation) is shown and the user is prompted to rate how well they remembered (right)

3 Comparing candidates methods for the sentence generation component

This section describes our simulated study to narrow down the methods and configurations that could optimize the system's objective to two that can be tested in the user study.

3.1 System objectives

The system's objective is to suggest sentences ("tasks") for the user to review, while following as closely as possible the due dates of the contained words coming from the spaced repetition scheduler. This results in the following three main objectives imposed by the first research question:

1. Maximize the correctness of the sentence
2. Maximize the amount of due and future due words contained, prioritize by upcoming due dates
3. Avoid sentences exceeding ten words (which was the maximum length that three test users reported not finding overwhelming)

3.2 Simulated Metrics

To automatically evaluate the different methods, we simulated their use over 20 days by a user who remembers any word with an 85% chance and then calculated the following automated metrics:

1. A scheduling score measuring how well the spaced repetition scheduling is adhered to and only due and future due vocabulary is used (for more details on the scheduler, see 4.1)
2. Too long sentences, to measure the fraction of sentences that are longer than the ten word limit from the third objective

We defined the scheduling score as the average fraction of the scheduling intervals wasted by scheduling words before they are due or 1 when a new word is introduced without the user asking for it, to discourage exponential vocabulary growth. It can be between zero and one and should be minimized.

$$S = \frac{1}{n_{tasks}} \sum_{tasks} \frac{1}{n_{taskwords}} \sum_{taskwords} s_{word}$$

$$s_{word} = \begin{cases} \frac{max(t_{due} - t_{now},\ 0)}{t_{due} - t_{last\_seen}} & \text{if in user vocab} \\ 0 & \text{user requested new word} \\ 1 & \text{new word, not requested} \end{cases}$$

Additionally, the correctness of the sample sentences was rated by a human evaluator and GPT-3.5-turbo-0301. While the human saw 20 samples per method, the LM saw 1000. They agreed fairly (Cohen's Kappa = 0.35), indicating that the LM's ratings can be useful when based on larger samples, but should not solely be relied upon.

3.3 Sentence Generation Methods

We implemented a variety of methods for generating or selecting sentences for testing purposes. Reinforcement learning with a static reward function (scheduling score) and modifying the probability distribution of a PLM directly (GPT-2 and OPT-1.3B) were briefly explored but were not able to generate at least 50% correct sentences that contained at least one of the words it was given as inputs. Meanwhile, retrieval of suitable sentences from a corpus and few-shot prompting did pass and they were thus moved on to the next stage where we subjected different configurations to the previously listed metrics.

The BM25 retrieval algorithm (Robertson and Zaragoza, 2009) was taken as a starting point for the retrieval method. It is suitable insofar as it ranks the sentences based on how many of the query words they contain and gives reduced importance the more common a query word is. We modified BM25 to add query word weights to give a higher importance to words that are due earlier (e.g. a word due today gets a higher weight than a word due tomorrow). We discount query words with exponential decay the longer in the future they were due. The following formula was used to rank the sentences: BM25(query, sentence) =

$$\sum_{\mathrm{w} \in \mathrm{query}} \left(\mathrm{idf_w} \frac{(\mathrm{k1}+1) \cdot \mathrm{q\_freq_w}}{\mathrm{q\_freq_w} + \mathrm{k1}(1 - \mathrm{b} + \mathrm{b}\frac{\mathrm{sent\_len}}{\mathrm{avgsl}})(\mathrm{dtd_w}+1)} \right)$$

Where $\mathrm{idf_w}$, $\mathrm{q\_freq}_w$, sent_len, avgsl as in BM25,
dtd means days until the word is due for repetition,
$\mathrm{k1} = 1.5$ and $\mathrm{b} = 0.75$

Same-day repetitions of a task are disallowed by finding the best-ranking sentence that had not been previously shown. In addition to this standard version described above, we test a version that selects the task with the best scheduling score among the 25 best-ranking tasks. We chose the Wiki-40B Corpus (Guo et al., 2020) as the source of the sentences since it is one of the largest corpora for Danish (and 40+ languages in total, allowing for easy adaption, even though the BM25 would have to be re-tuned for some languages' features, e.g. different tokenization) with ca. 200MB worth of Danish sentences and, as it is sourced from Wikipedia articles, contains mostly correct use of the language. We removed sentences with rare words (not in the 25000 most frequent from the language), shorter than two, or longer than 10 words. After the filtering, the resulting corpus contained 64259 sentences, of which the average length was 5.9 words.

For the prompting approach, we chose GPT-3.5-turbo-0301 as the language model since it was the largest model that was partly trained on Danish data (0.1%, 220 million words in Danish (Brown et al., 2020)) at the time of writing, trained to be helpful with answering prompts containing instructions and relatively cheap to use. We explore different zero and few-shot prompts, with the best performing one given in appendix A and used for all further experiments. Input words are taken from the words scheduled for the current day and upcoming ones if fewer words were due on the day than the method takes as input. We also test two different system messages given to the model before the prompt, instructing it to generate a maximum of 5 words in the first and 10 words in a correct and meaningful sentence in the second. We also explore two temperature settings (0.2 and 0.8), five versus ten

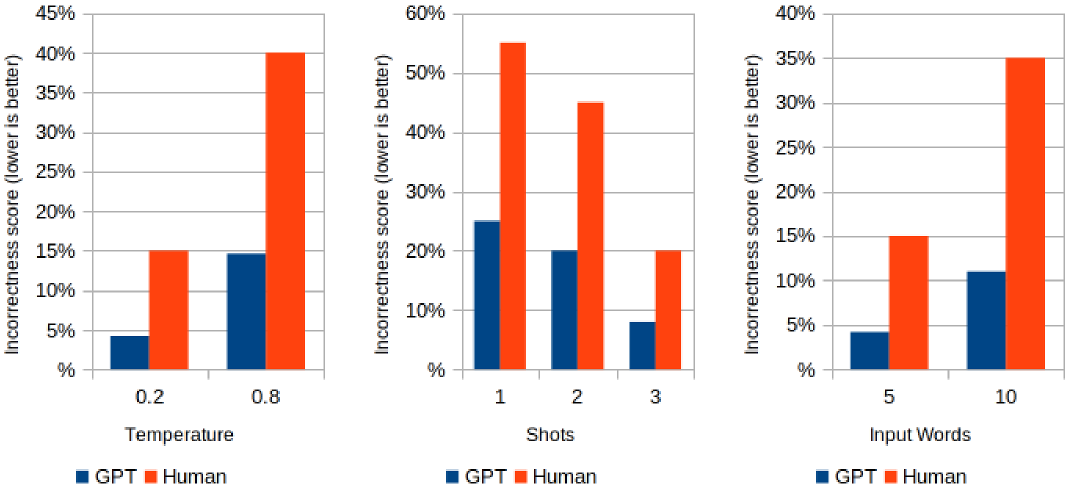


Figure 2: Influence of different temperatures, number of shots, and number of input words on correctness

input words, and one-, two- and three-shot prompting. A zero-shot approach resulted too often in the word list just being returned verbatim, so it was not further pursued. Similarly to the retrieval method, the approach of selecting the output with the best scheduling score out of three generations was implemented. Returning three generations also allowed us to filter out incorrect ones by prompting GPT-3.5 about their correctness before selecting the best. Not all combinations of these configurations were tested, but only one factor was altered at a time.

We also explore a hybrid method choosing BM25 retrieval and GPT-3.5 each with a 50% chance.

A sample of the outputs of different methods for different inputs is given in appendix B.

3.4 Results of Simulated Metrics

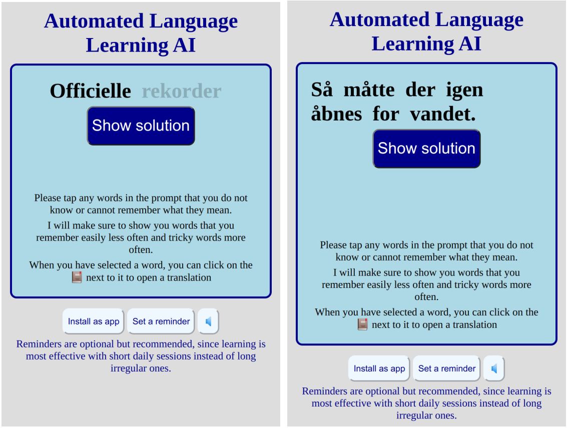
One of the biggest issues with GPT-3.5 for generating tasks was a tendency to loop because of lemmatization or the lack thereof. Above all, it is a pedagogical question whether the user's vocabulary should consist only of the lemmas the user has seen or all the different forms of these lemmas independently, and the answer arguably depends on how morphologically rich the language is. For simplicity, in this work, it was decided to treat all forms of a lemma separately since the other approach would require using a lemmatizer on the generated tasks, and with the best Danish lemmatizer at the time of writing having an accuracy of just 0.95, incorrect lemmas would make it into the vocabulary.

With the previously chosen prompt and parameters, GPT-3.5 tends to generate the word form related to the input word, which best fits the grammar of the sentence, possibly due to not "thinking ahead" when it starts the sentence, even when a sentence "Generate the exact words forms given" was added to the prompt. This tendency leads to another form being reviewed than is due, while the due form remains due, thus leading to it being generated again in the next task, possibly going on forever.

The retrieval and the hybrid method did not suffer from this problem, since the retrieval method uses exact matches. The hybrid model could temporarily fall into a loop when using the LM method but would eliminate the troublesome word from the due words as soon as it uses the retrieval method, which it does 50% of the time.

All the different combinations of configurations tested and their scores on the metrics are given in appendix C. Figure 2 visualizes the influence of different parameters on the correctness.

Overall, the scheduling scores are very good, meaning that most words in the tasks must have been due on the exact day they were generated. The fact that most scheduling scores are below 0.1 means that on average, less than one in ten words in the tasks were out-of-user-vocabulary, and less than one in five was not due on the day the sentence was generated. Most sentences the best GPT method generated were correct, however, the user would see a substantial amount of wrong grammar or nonsense (around 15% according to the human evaluator), impacting learning outcomes and possibly motivation. The hybrid method was rated 10%



Automated Language Learning AI

Officielle rekorder

Please tap any words in the prompt that you do not know or cannot remember what they mean.

I will make sure to show you words that you remember easily less often and tricky words more often.

When you have selected a word, you can click on the 📕 next to it to open a translation

Reminders are optional but recommended, since learning is most effective with short daily sessions instead of long irregular ones.

Automated Language Learning AI

Så måtte der igen åbnes for vandet.

Please tap any words in the prompt that you do not know or cannot remember what they mean.

I will make sure to show you words that you remember easily less often and tricky words more often.

When you have selected a word, you can click on the 📕 next to it to open a translation

Reminders are optional but recommended, since learning is most effective with short daily sessions instead of long irregular ones.

Figure 3: Screenshots of a task as seen by baseline (left) and retrieval/hybrid group (right)

incorrect, which is better but still high.

For the GPT-3.5 models, we found that using a low temperature parameter of 0.2, five input words, three shots, a system message instructing the model to generate up to ten output words, and selecting the output with the best scheduling score out of three generated outputs, where outputs rated by the model itself as incorrect when asked again are discarded, gave the best results. It was the most correct out of the variants tested, was tied for the best scheduling score, and had an acceptable amount of sentences that were longer than the goal of ten words. Thus, it was decided to use this configuration in the hybrid model. When it comes to the BM25 models, unsurprisingly all of them were rated 100% correct. Using the best-out-of-25 strategy improved the scheduling score and had no other downsides, and was thus chosen as the retrieval method to test in the user study and to be part of the hybrid model. As was to be expected with the hybrid model using two models 50% of the time each, most metrics come in right between the used GPT-3.5 model and the used BM25 model. Thus, solving the looping problems and performing decently in the metrics, it was decided that the hybrid model is adequate to be the way how LM generated tasks are tested in the user study. No purely LM-based model was selected since the looping problem would have too big an impact on the user experience.

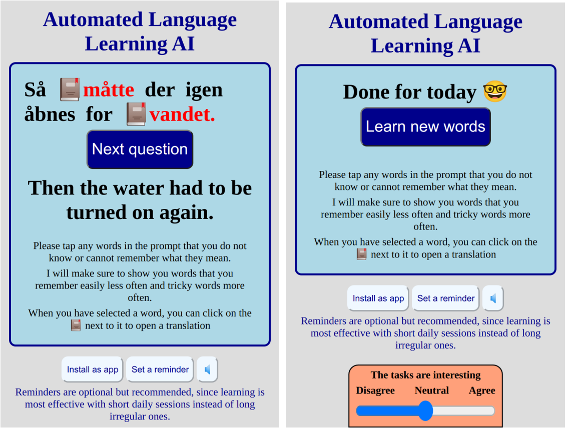
4 User study

In addition to the two selected methods, a baseline method was developed to allow for comparison to the proposed methods in the user study. As the baseline, it was chosen to associate a set sentence (the one with the best BM25 score) with each word in the vocabulary, which is then shown when the word is due. The due word is specially marked and only it can be reported as remembered correctly or not for the spaced repetition. This mimics the common approach of putting a single word on the spaced repetition flashcard, accompanied by some example sentences, as identified in section 2.1, but is put into a comparable format to how the two selected methods are presented to the user.

4.1 Test system design

For the user study, a progressive web app was developed as a front-end for the user to interact with the generated tasks. Upon opening the app, a user would see the first generated task (figure 3). After thinking about a translation to the task, they click a button to show the solution. They would then mark all words in the task that they did not remember correctly (or had never seen before). Through seeing a solution and the option to click a dictionary icon next to the words they marked, they could learn the meaning of new words, and refresh their memory of old ones. This is shown in figure 4 on the left. After selecting all unknown words, they would press the button again to be shown the next task, and so on, until they either wanted to stop, or they had reviewed all words that, according to the spaced repetition system, were due on the day. At that time, a "done for today" screen was shown, as seen in figure 4 on the right. This was intended as a natural stopping point for users, however, if they were motivated enough to spend more time, they were given the option to add five new words to the vocabulary and the system would generate tasks containing these words and show them immediately. This option could be used repetitively, so the user could study for as long as they wanted. To schedule the spaced repetition, the SM-2 algorithm (Wozniak, 1990) was chosen, a variation of which is for example used by Anki (Elmes), one of the most widely used spaced repetition programs. One simplifying modification was made: While the SM-2 algorithm grades responses on a six-point scale to express how difficult it was to recall the information, a two-grade scale was used, corresponding to grades 1 (not recalled) and 4 (recalled correctly) in the original SM-2 algorithm.

Whenever the user requested to learn new words (beyond those that the retrieval and hybrid method would generate by accident in the sentences), five new words were added to the vocabulary starting



Så 📕måtte der igen åbnes for 📕vandet.

Then the water had to be turned on again.

Please tap any words in the prompt that you do not know or cannot remember what they mean.

I will make sure to show you words that you remember easily less often and tricky words more often.

When you have selected a word, you can click on the 📕 next to it to open a translation

Reminders are optional but recommended, since learning is most effective with short daily sessions instead of long irregular ones.

Done for today 🤓

Please tap any words in the prompt that you do not know or cannot remember what they mean.

I will make sure to show you words that you remember easily less often and tricky words more often.

When you have selected a word, you can click on the 📕 next to it to open a translation

Reminders are optional but recommended, since learning is most effective with short daily sessions instead of long irregular ones.

The tasks are interesting

Figure 4: Screenshots of solution being shown with two words selected as unknown (left), and "done for today" screen (right) with the interestingness prompt being shown, as described in section 4.2

at the most frequent ones in the language, using the WordFreq (Speer, 2022) frequency list Python module.

4.2 User study setup and metrics

26 test users were recruited for the user study, mainly through social media from the researchers' acquaintances. The only exclusion criterion was that the user should not be completely fluent in Danish. The test users can thus not be assumed to be representative of the general population. Participants were shown an initial questionnaire, collecting demographical information and their background in language learning and initial motivation, which were treated as potential confounding variables. Participants were aged 19 to 56 (mean 28.9, std 11.1). 9 were female and 17 male and they had 15 different native languages. 17 were living in Denmark and 9 had never lived there. Those in Denmark had lived there from ten months up to 6 years (mean 2.5 years, std 1.4 years). 14 had learned Danish before and out of them, 10 of these had used the language outside of a class context. 23 had previously used other language-learning apps. Users reported an average motivation of 3.1 on a 1-5 scale, std 1.0) and mainly career prospects, curiosity, and social life as the motivating factors.

The participants were allocated randomly into the three intervention groups using blocked randomization, the two blocks being those who previously had learned Danish and those who had not. The study was double-blind, except that the tasks were presented with only one word highlighted to the baseline group. This means that if two participants compared, they could find out about not being in the same group, but not whether they were in the treatment or control group. It lasted ten days, during which users were allowed to choose freely, how much time they would like to spend using the app. The following metrics were collected either from usage data or questionnaires to assess learning outcomes and user engagement:

1. User vocabulary growth (words remembered minus words known at first exposure)
2. Time efficiency (words remembered / minute spent)
3. Word effectiveness (new words remembered / words seen)
4. Number of distinct words seen
5. Total time using the system
6. User's self-reported interestingness, enjoyment, perceived learning, challengingness, and confusion at random points while learning, prompt shown in figure 4 on the right

The data was analyzed for correlations between all the metrics and demographical data, in case these uncovered some major confounding factors. For the significance testing, the one-sided Mann–Whitney U test (Mann and Whitney, 1947) was used to determine the significance of the differences between the groups with regard to the metrics. It tests whether a probability distribution is greater than the other and does not assume normally distributed data. Results were considered significant if the p-value was smaller than 0.05.

4.3 Results of User Study and Discussion

During the user study, the single-word group only saw 98 different tasks, the retrieval group saw 319, and the hybrid group had 400 distinct tasks. Differences were mainly observed in total vocabulary growth, efficiency (figure 5), and enjoyment. Please see appendix D for a table and figures of the main results. The users' vocabulary grew by a 7 word median but with a high standard deviation of 19.3. Both the group using a language model and the pure retrieval group achieved around four-fold greater time efficiency of their vocabulary growth than the single-word group, while seeing three times more words and four-to-six times higher overall vocabulary growth, even though the latter was not significant for the hybrid group. In all of the user-reported metrics related to engagement, the intervention groups fared slightly better than the single-word baseline, but the difference was

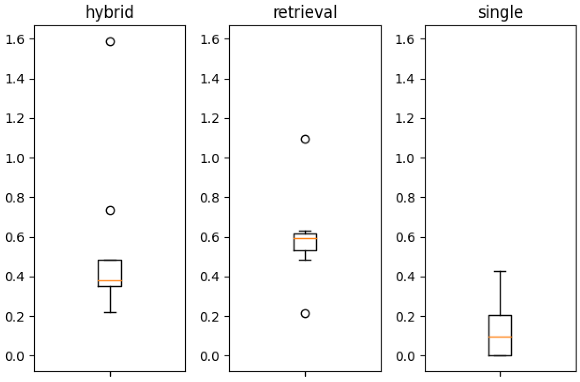


Figure 5: Box plot of the efficiency (vocabulary growth per minute) in the different groups

only significant for enjoyment, where the retrieval group had significantly higher ratings than hybrid (p=.028) and baseline group (p=.042). Most users in this group reported enjoying using the app.

These results indicate that, compared to single-word spaced repetition with set assigned sentences, generating or selecting dynamic sentences based on multiple due words, can indeed increase learning outcomes and user engagement. It seems likely that using sentence-based spaced repetition first and foremost manages to show users more new words to learn in less time, especially for beginners (Negative correlation Pearson's $r = -0.4$ between vocab growth and previous knowledge). This increases efficiency and vocabulary growth since users still retain the same fraction of words seen or even slightly more when they focus on several words in the sentence and see words in various contexts. The increased efficiency then probably leads to higher enjoyment (Pearson's $r = 0.5$ between efficiency and enjoyment).

The differences between the two intervention groups have mostly been minor. Still, they were significant for enjoyment and almost significant for efficiency, which could have led to the increased enjoyment.

5 Implications

The results mean first and foremost, that using a sentence-based spaced repetition scheme should be preferred over using single-word spaced repetition, even when the single word is shown in the context of an example sentence. This will show users more vocabulary in less time, increasing efficiency and thus enjoyment.

Since a retrieval model is far less costly in terms of computing costs and there is light evidence that it is the more time-efficient and enjoyable option, it could be advisable to prefer retrieval over LM-based options, but this would have to be proven in a bigger trial to achieve significant results after Bonferroni corrections (see limitation in section 7).

On the other hand, even though this specific prompting-based LM method and configuration could not outperform retrieval, with the current rapid advancements in LM size and tasks they can perform through prompting, other LMs e.g. GPT-4, which has substantially more parameters than GPT-3.5, could improve correctness and possibly number of due words in the prompt.

While our experiments compared the proposed system to a conventional baseline under similar conditions and presentation, we can also compare the results to previous literature. Thorndike (1908) studies learners' efficiency of learning lists of word pairs and mentions an average of 0.57 words per minute, with 0.34 recalled words after 42 days. Thus, it seems that hybrid and retrieval groups with mean of 0.54 and 0.6 words per minute recalled after a few days had a higher efficiency than the results from Thorndike's study, even though not directly comparable, since Thorndike's study did not have the problem of time being wasted on previously known words, which we did not count for vocabulary growth.

6 Conclusion

The aims of this work were first to identify NLP paradigms and configurations for sentence generation that can optimize spaced repetition timing and best avoid out-of-user-vocabulary words, while keeping the correctness of the generated sentences as high as possible, and then to quantify these methods' influence on user engagement and learning outcomes among language learners, compared to conventional approaches.

Two methods of achieving these goals were developed: one based on retrieval of suitable sentences from a corpus of high-quality sentences using many upcoming due words as queries, and the other was few-shot-prompting a PLM to generate sentences from a subset of the due words. Both methods were found to be able to form sentences comprised mostly of words from the user vocabulary, soon to be due and mostly correct, thereby reaching the objectives. While the retrieval method reached 100% correctness, the LM method optimized the spaced repetition scheduling even better but had worse correctness and had an unsolved problem with looping due to the treatment of lemmas, despite multiple countermeasures, making it unsuitable for deployment to users. A hybrid method switching between retrieval and LM generation could solve the looping problem while optimizing the research question's objectives.

Consequently, the hybrid and the retrieval method were compared to a baseline to answer the second research aim. It was found that the proposed sentence-based spaced repetition significantly increased learning outcomes (four-to-six-fold) compared to the baseline, primarily by increasing efficiency and vocabulary growth by showing more words more quickly, without decreasing the fraction of words remembered by learners. In the retrieval group, a significantly higher enjoyment was observed, possibly due to the higher efficiency, hinting at a higher user engagement.

It can thus be concluded that it is beneficial to use the proposed sentence-based spaced repetition over the conventional approach and that the retrieval approach might be advisable over LM-based or hybrid approaches, but that a bigger trial comparing the two is necessary, and further developments, such as fixing problems with lemmatization and looping and higher correctness possibly achievable with newer language models could improve the results when using a more advanced LM based method in the future.

7 Limitations

Convenience sampling has been employed to choose study participants. Participants were very diverse in some aspects such as native language, but very homogeneous in others, such as previous usage of language learning apps. This means that participants are not representative of the general population. While it can be reasonably assumed that learning works similarly in all humans, the evidence for the effect observed is strongest for people similar to the participants. It might not be generalizable to persons with completely different backgrounds, for example school children, a large sub-group of language learners.

The recruitment through acquaintances could affect the user-reported metrics through the social desirability bias, making participants more likely to give more favorable ratings. This has been partly mitigated by emphasizing the anonymity of the participants' answers, but it cannot fully be avoided. However, it affects all test groups equally, since users did not know which intervention they had been assigned to, so the results remain comparable between the groups.

Furthermore, the sample size was small with 26 participants, looking at a population of hundreds of thousands of Danish learners or possibly billions of persons learning languages in general. This sample size might not have been big enough to detect some possible differences between the hybrid group and the control group or the retrieval group and the hybrid group. It was, however, big enough, to detect some of the most pronounced effects that this work tried to assess.

The user study analyzed the differences between three groups in eleven metrics for significance using a 0.05 p-value threshold. The large number of comparisons makes false positives more likely to occur. While it can be assumed that the majority of differences reported as significant are indeed significant, it should be noted that the use of Bonferroni correction, to reduce the total possibility of having any false positives to 0.05, would only leave the difference between the efficiency of the retrieval vs single group as significant.

The duration of the user study of ten days also only allows for drawing direct conclusions for short-term use, but, this was tried to be mitigated by measuring engagement as a possible predictor of long-term learning outcomes.

The choice of Danish as the language for the user study is a slight limiting factor for generalizability. While it is reasonable to assume that learning happens in a similar way and is influenced by similar factors in most languages, details about the language such as its morphology, e.g. having many word forms for each lemma, could lead to reduced or increased suitability of the proposed approach and possibly increased importance of storing user vocabulary as lemmas instead of word forms.

References

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*, pages 819–826, Vancouver, British Columbia, Canada. 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. ArXiv:2005.14165 [cs].

Shana K. Carpenter, Nicholas J. Cepeda, Doug Rohrer, Sean H. K. Kang, and Harold Pashler. 2012. Using Spacing to Enhance Diverse Forms of Learning: Review of Recent Research and Implications for Instruction. *Educational Psychology Review*, 24(3):369–378.

Diogo Cruz. 2023. Creating Flashcards with LLMs.

Damien Elmes. Anki.

Noureen Fatima, Ali Shariq Imran, Zenun Kastrati, Sher Muhammad Daudpota, and Abdullah Soomro. 2022. A Systematic Literature Review on Text Generation Using Deep Neural Network Models. *IEEE Access*, 10:53490–53503.

Alexej Gossmann. 2024. Comparing GPT-4, 3.5, and some offline local LLMs at the task of generating flashcards for spaced repetition (e.g., Anki).

Mandy Guo, Zihang Dai, Denny Vrandečić, and Rami Al-Rfou. 2020. Wiki-40B: Multilingual Language Model Dataset. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2440–2452, Marseille, France. European Language Resources Association.

Tao Hao, Zhe Wang, and Yuliya Ardasheva. 2021. Technology-Assisted Vocabulary Learning for EFL Learners: A Meta-Analysis. *Journal of Research on Educational Effectiveness*, 14(3):645–667. Publisher: Routledge _eprint: https://doi.org/10.1080/19345747.2021.1917028.

Jakub Jankowski. 1999. Effective learning: Twenty rules of formulating knowledge.

Daniel Jurafsky and James H. Martin. 2023. *Speech and Language Processing*.

Sebastian Leitner. 1972. *So lernt man lernen. Angewandte Lernpsychologie – ein Weg zum Erfolg*. Verlag Herder, Freiburg im Breisgau, Germany.

H. B. Mann and D. R. Whitney. 1947. On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *The Annals of Mathematical Statistics*, 18(1):50–60. Publisher: Institute of Mathematical Statistics.

Yuki Okano, Kotaro Funakoshi, Ryo Nagata, and Manabu Okumura. 2023. Generating Dialog Responses with Specified Grammatical Items for Second Language Learning. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 184–194, Toronto, Canada. Association for Computational Linguistics.

Restrepo Ramos and Falcon Dario. 2015. Incidental Vocabulary Learning in Second Language Acquisition: A Literature Review. *Profile Issues in Teachers‘ Professional Development*, 17(1):157–166. Publisher: Universidad Nacional de Colombia.

Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*, 3:333–389.

Robyn Speer. 2022. rspeer/wordfreq: v3.0.

Edward L. Thorndike. 1908. Memory for paired associates. *Psychological Review*, 15(2):122–138. Place: US Publisher: The Review Publishing Company.

Maarten van der Velde. 2023. Flashcard Fundamentals #3: Generating Flashcards using AI.

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. ArXiv:2206.07682 [cs].

P. A. Wozniak. 1990. *Optimization of learning: A new approach and computer application*. Ph.D. thesis.

Piotr Wozniak. SuperMemo.

Michael Zock, Reinhard Rapp, and Chu-Ren Huang, editors. 2014. *Proceedings of the 4th Workshop on Cognitive Aspects of the Lexicon (CogALex)*. Association for Computational Linguistics and Dublin City University, Dublin, Ireland.

Fidel Çakmak, Ehsan Namaziandost, and Tribhuwan Kumar. 2021. CALL-Enhanced L2 Vocabulary Learning: Using Spaced Exposure through CALL to Enhance L2 Vocabulary Retention. *Education Research International*, 2021.

A Few shot prompt

The following few-shot prompt was selected as it was the best performing of several variations tried:

```
Lav en korrekt sætning med de givne ord.

###

ord: en har at sådan; sætning: Vi har ønsket, at der var en løsning.
ord: nyhed for god rimmelig; sætning: Det er en god nyhed for os!
ord: rigtigt se hellere i udenfor københavn; sætning: Jeg vil hellere
     kunne se rigtigt udenfor.
ord: [List of 5/10 due words]; sætning:
```

Figure 6: Three shot prompt (First line translates to "Make a sentence with the given words". "ord" translates to "words", "sætning" to "sentence".)
The one and two shot version only used the first or first two of these examples.

B Output samples of each method

Method	Input Words	Output Sentence
single	**det**	**Det** er **det** ikke.
single	trygt	I mellemtiden havde Wilhelm været i Rom.
retrieval	i, og, **er**, af, det, at, **en**, til, på, **jeg**	**Jeg er en** mand.
retrieval	trygt, udland, undre, **er**, deltage, hun, zone, forsøger, **dannede**, **ét**, **kemisk**, træk, typer, tyst, ulovlig, klage, på, mio, **det**, retten, også, manager, general, tavs, forgæves, samfundet, party, præsidenten, højesteret, spurgt, derpå, af, overvejelser, episk, privatliv, historiske, beskyttelse, danskerne, tegnede, ting, som, udgang, markedsføring, ledsaget, de, blå, brikker, en, jeg, mand, rejste, rose, mary, 2, nu, lider, mini, israel, willie, derfor, vi, coffee, grund, **stof**, fikset, medlemskab, o, airways, british, for, hjørring, mørkt, der, ud, henrettet, til, stk, køber, blev, i, little, viden, at, og	**Det dannede stof er ét kemisk stof**.
gpt3.5	**en**, **er**, af, **på**, **jeg**	**Jeg er på en** mission.
gpt3.5	**trygt**, *udland*, *undre*, **er**, **deltage**	Jeg *undre*r mig over, om det **er trygt** at **deltage** i aktiviteter i *udland*et.

Table 1: Word Lists and Sentences for each of the three selected methods, first for a new user, then after a few iterations of studying. Input words used in the output are in bold, or in italic if not the exact form but the same lemma.
In line 2, there was no sentence in the corpus containing this word form. In line 4, the exact same sentence had been generated on a previous day.

C Results of simulated model evaluation

Model	Tempe-rature	Input Words	Shots	System Message	Best out of n, critera	Sched score	>10 words	Incorrect (GPT I Human)
gpt3.5	0.2	5	3	none	3, best sched score	0.068	18.7%	8.5% I 50%
gpt3.5	0.2	5	3	1	3, best sched score	0.124	5.4%	11.5% I 25%
gpt3.5	0.2	5	1	2	3, best sched score	0.094	7.0%	25.3% I 55%
gpt3.5	0.2	5	2	2	3, best sched score	0.068	12.7%	20.2% I 45%
gpt3.5	0.2	5	3	2	3, best sched score	0.070	19.1%	8.0% I 20%
gpt3.5	0.2	5	3	2	3, prefer correct ->best sched score	0.068	19.6%	4.2% I 15%
gpt3.5	0.8	5	3	2	3, prefer correct ->best sched score	0.082	13.1%	14.6% I 40%
gpt3.5	0.2	10	3	2	3, prefer correct ->best sched score	0.077	44.1%	11.0% I 35%
BM25	-	25	-	-	1	0.113	9.9%	0% I 0%
BM25	-	25	-	-	25, best sched score	0.098	8.5%	0% I 0%
Hybrid	0.2	5 (LM) / 25 (BM25)	3	2	3 (LM) / 25 (BM25), prefer correct -> best sched score	0.078	11.2%	4.5% I 10%

Table 2: Comparison of the considered models' and parameters' scores on the metrics.
System messages:
1: "Du er conciseGPT, dine svar er meget korte, maks 5 ord.",
2: "Du er conciseGPT, dine svar er meget korte, maks 10 ord, men korrekte og giver mening."
The column "Best out of n, criteria" describes how many outputs were generated by the method and the criteria by which the best was selected as the final output. "Prefer correct" means that out of the n results, only the correct ones (determined by prompting GPT-3.5) were considered for the next criterion. If none was correct, all were considered.

D Results of user study

Method		Vocabulary Growth	Time Efficiency (words/min)	Word Effectiveness	Words Seen	Total Time Spent (min)
Overall	Median	7	0.38	0.12	46.5	17.4
	Mean	11.5	0.43	0.15	65.3	23.7
	Std	19.3	0.35	0.13	82.0	27.5
Single Word	Median	1.5	0.10	0.05	15.0	16.4
	Mean	3.4	0.14	0.12	24.0	21.9
	Std	4.1	0.16	0.15	19.5	25.0
Hybrid	Median	6.0	0.38	0.12	55.0	17.1
	Mean	18.8	0.54	0.16	78.0	27.3
	Std	31.0	0.42	0.14	82.4	39.3
Retrieval	Median	10.0	0.59	0.17	48.0	26.2
	Mean	11.4	0.60	0.18	89.4	21.7
	Std	7.7	0.24	0.12	106.6	15.9
p-value	hybrid $\leq$ single	0.056	0.003		0.005	
	retrieval $\leq$ single	0.017	0.001		0.034	
	retrieval $\leq$ hybrid		0.089			

Table 3: Results of the measured metrics of the user study (p-values only shown if <0.1

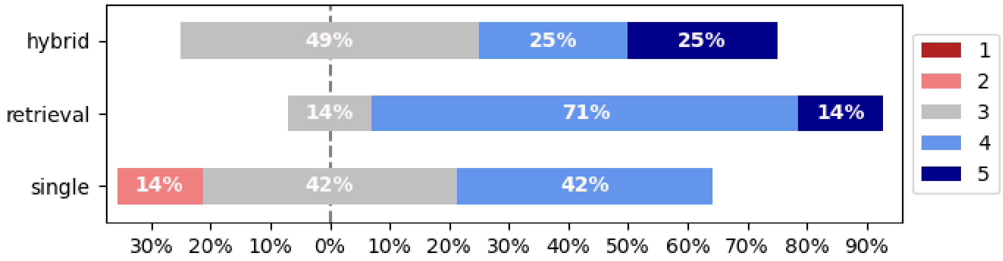


Figure 7: User ratings of "This is interesting" across the different groups (1 = disagree, 5 = agree)

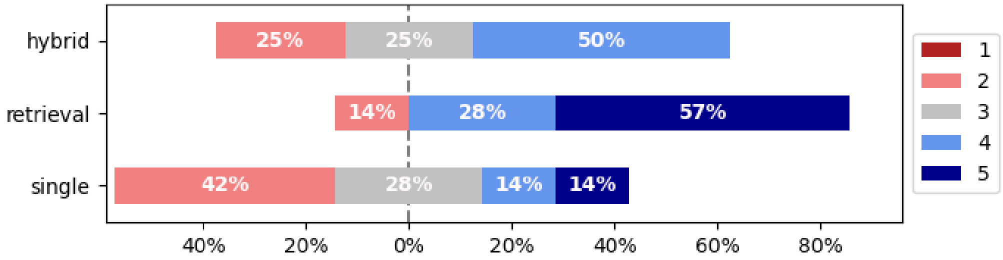


Figure 8: User ratings of "I am enjoying this" across the different groups (1 = disagree, 5 = agree)

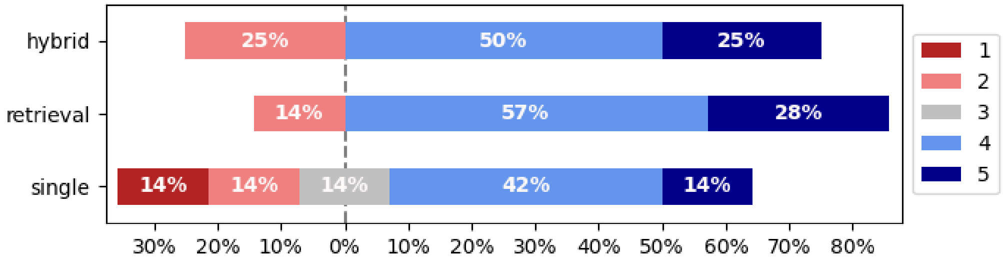


Figure 9: User ratings of "I am learning a lot" across the different groups (1 = disagree, 5 = agree)

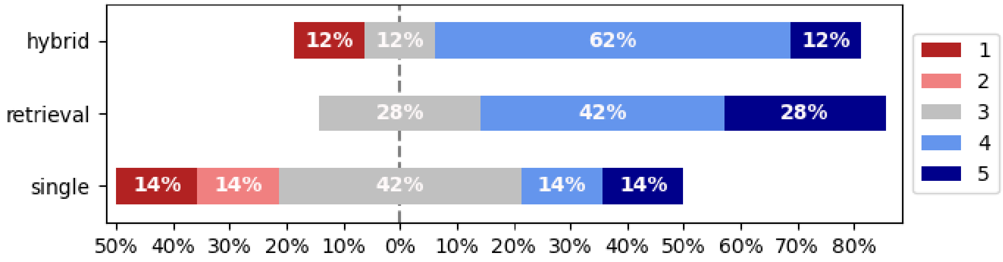


Figure 10: User ratings of "This is challenging" across the different groups (1 = disagree, 5 = agree)

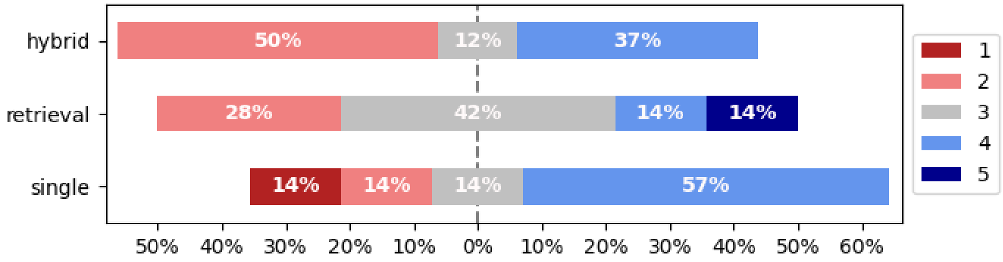


Figure 11: User ratings of "I am confused" across the different groups (1 = disagree, 5 = agree)

Using Large Language Models to Assess Young Students' Writing Revisions

Tianwen Li[1], **Zhexiong Liu**[2], **Lindsay Clare Matsumura**[1], **Elaine Lin Wang**[3]
Diane Litman[1,2], **Richard Correnti**[1]
[1]Learning Research and Development Center, University of Pittsburgh
[2]Department of Computer Science, University of Pittsburgh
[3]RAND Corporation, Pittsburgh, PA 15260 USA
{tianwen.li,lclare,dlitman,rcorrent}@pitt.edu
zhexiong@cs.pitt.edu, ewang@rand.org

Abstract

Although effective revision is a crucial component of writing instruction, few automated writing evaluation (AWE) systems specifically focus on the quality of the revisions students undertake. In this study, we investigate the use of a large language model (GPT-4) with Chain-of-Thought (CoT) prompting for assessing the quality of young students' essay revisions aligned with the automated feedback messages they received. Results indicate that GPT-4 has significant potential for evaluating revision quality, particularly when detailed rubrics are included that describe common revision patterns shown by young writers. However, the addition of CoT prompting did not significantly improve performance. Further examination of GPT-4's scoring performance across various levels of student writing proficiency revealed variable agreement with human ratings. The implications for improving AWE systems focusing on young students are discussed.

1 Introduction

The ability to write is foundational to academic success. Yet, national assessments show that nearly three-quarters of students in the United States are not proficient writers (NCES, 2012). A well-recognized approach for improving students' writing skills is to engage students in cycles of revising their essays in response to formative feedback (Graham and Perin, 2007; Graham and Sandmel, 2011). However, students rarely receive substantive formative feedback on their writing for multiple reasons. First, teachers can be reluctant to assign writing tasks that require students to work across drafts because providing formative feedback is time-consuming (Graham et al., 2014). Second, teachers can feel unsure about how to provide feedback to improve students' essay quality (Brindle et al., 2016). Finally, research shows that teachers are inconsistent in their feedback practices, and tend to focus on surface-level features of students' writing rather than the content of students' ideas and reasoning (Matsumura et al., 2002, 2023).

Automated Writing Evaluation (AWE) systems are gaining prominence as one approach to increasing students' opportunity to receive formative feedback. While research suggests that teachers generally respond positively to AWE systems and can see them as helpful time savers (Grimes and Warschauer, 2010; Palermo and Thomson, 2018), evidence is modest that AWE systems improve the quality of students' writing in the elementary and secondary grades (Graham et al., 2015). One reason why students' writing may not improve in response to automated feedback is that they often lack the skills necessary for effective revision (Roscoe et al., 2013; Wang et al., 2020). Wang et al. (2020) found that only 18% of students successfully implemented the feedback messages they received from an AWE system. For example, when asked to provide more evidence for their claims, students commonly repeated the examples that they had cited before. This highlights the importance of providing students with feedback that builds their revision skills, in addition to feedback that improves their writing quality.

Given that formative assessment fosters writing skill development by establishing and reinforcing clear criteria for successful writing (Matsumura et al., 2023), it is notable that few assessments target students' revision skills. Building on the previous discussion about the necessity of teaching students how to revise, we believe that formative assessments that precisely establish the criteria for effective revision can provide information to students and teachers about the extent to which

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 365–380
June 20, 2024 ©2024 Association for Computational Linguistics

revision goals are met and offer guidance for implementing revision feedback. To address this gap, our team developed a rubric for holistically assessing revision quality (Wang et al., 2020). By ‘revision quality’, we specifically examine whether revisions students made were aligned with the feedback provided, and the extent to which it improved the essay with respect to evidence use. This is in contrast to revisions that may improve essay quality in ways not aligned with the content of feedback messages.

In the context of AWE systems, automatically assessing the revision process is a necessary area of development. Most systems have focused on assessing overall improvement in essay quality. Although these systems can detect revisions, they tend to assign scores or provide feedback based on the overall essay quality, rather than attend to the quality of the revisions undertaken (Foltz and Rosenstein, 2017; Mayfield and Butler, 2018). Recent advancements in large language models (LLMs) show significant promise for analyzing and evaluating student revision quality. GPT-4, standing out among these models, specifically has been shown to generate scores that are comparable to those given by human evaluators (Mizumoto and Eguchi, 2023; Naismith et al., 2023; Tate et al., 2023; Xia et al., 2024; Xiao et al., 2024). While most of these studies have concentrated on GPT-4’s ability to assess writing quality, our study extends previous research by investigating the effectiveness of GPT-4 for evaluating revision quality with different prompting strategies. Given that students often find essay revision challenging, it is essential to provide a revision score that reflects diverse revision patterns. This study represents an initial step in exploring GPT-4’s capability to score revisions, setting the stage for offering personalized feedback on students’ revision practices in future research.

In this study, we specifically explore GPT-4’s performance in assessing the revision attempts of young students (ages 10 to 12) who often exhibit less structured and sophisticated writing styles. Given that most existing research concentrates on evaluating essays by adolescents and adults (e.g., Naismith et al., 2023; Xiao et al., 2024), it is of interest to explore how GPT-4 adapts to the writing of younger age groups. In addition, as students may display a wide range of writing proficiency, it is crucial to ensure that GPT-4 does not exhibit systematic biases that could compromise scoring accuracy.

Two research questions are addressed:

1. How accurately can GPT-4 assess the revision quality of students’ argumentative writing in comparison with human raters?
2. How does GPT-4’s performance in evaluating revisions vary across different levels of students’ argumentative writing abilities?

2 Data

In this section we describe the dataset of students’ essays, the rubric used for assessing students’ revision quality, and the process for evaluating these revisions by human raters.

2.1 RTA space dataset

The corpus for our investigation is drawn from a study of eRevise, an AWE system designed to improve students’ argumentative writing in the fifth and sixth grades (Correnti et al., 2022; Zhang et al., 2019). eRevise was designed to score responses and provide feedback to students on the Response-to-Text Assessment (RTA). The RTA aims to assess the quality of students’ ability to reason about texts in their writing and to use text evidence to support their claims (Correnti et al., 2012; Correnti et al., 2013). The form of the RTA used in this study is based on a non-fiction article about government funding for space exploration (RTA$_{Space}$). To administer the RTA, a teacher reads the text aloud to students as they follow along with their copy of the article. The teacher also poses planned questions at certain points in the articles and defines some vocabulary words to ensure that all students comprehend the article in advance of writing. Students respond to the following prompt:

> Consider the reasons given in the article for why we should and should not fund space exploration. Did the author convince you that “space exploration is desirable when there is so much that needs to be done on Earth”? Give reasons for your answer. Support your reasons with 3-4 pieces of evidence from the text.

After students submit their first drafts, the system uses NLP features generated during the automatic scoring of students’ initial essays (including the number of pieces of evidence, specificity of evidence, concentration of evidence, and word count) to select formative feedback on evidence. There

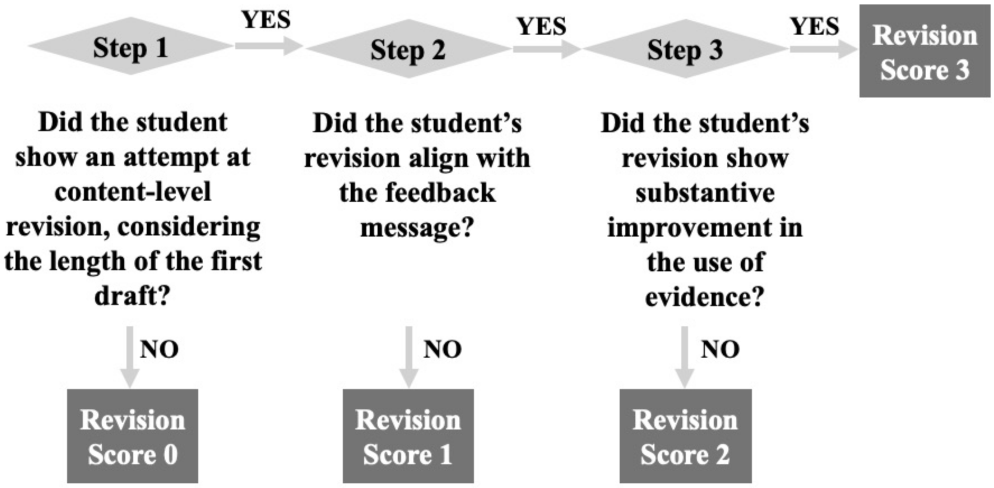

Step 1
YES
Step 2
YES
Step 3
YES
Revision Score 3
Did the student show an attempt at content-level revision, considering the length of the first draft?
Did the student's revision align with the feedback message?
Did the student's revision show substantive improvement in the use of evidence?
NO
NO
NO
Revision Score 0
Revision Score 1
Revision Score 2

Figure 1: Human rater evaluation steps

are three levels of feedback (Appendix A). Feedback Level 1 focuses on completeness (i.e., guides students to provide more evidence) and guides students to be more specific about the evidence they reference. Feedback Level 2 also directs students to be more specific, in addition to explaining their evidence. Finally, Feedback Level 3 guides students to explain their evidence and connect it to their overall argument (Correnti et al., 2020; Wang et al., 2020). After receiving the tailored feedback, students make revisions to their essays accordingly.

The RTA$_{Space}$ dataset contains a total of 600 essay pairs, which include both initial and revised essay drafts, collected from thirty-four fifth and sixth-grade ELA teachers in Louisiana who participated in the study during the 2018-2019 school year.

2.2 Human assessment of students' revision quality

Our team developed a holistic rubric to assess revision quality based on a detailed qualitative analysis of how fifth and sixth graders applied the automated feedback they received (Wang et al., 2020). We identified four levels of revision: 0 = No attempt at implementing feedback; 1 = Attempted to implement feedback, but no improvement in evidence use; 2 = Slight improvement in evidence use; 3 = Substantive improvement in evidence use. These four levels of revision were further transformed into a sequential flow of reasoning steps that guide human raters' scoring process (Figure 1). In addition, since initial drafts were categorized into three levels, each offering different focuses for revision, the ways in which students attempt to apply the feedback could vary. As a result, beyond the four abstract criteria used to assess the quality of revisions, the rubric was supplemented by specific, frequently observed patterns identified by human raters at each revision score (Appendix B).

For example, if a student receives Feedback Level 1 which focuses on the completeness and specificity of evidence, a successful revision (score 3) involves adding more than one new piece of evidence from the text that was not previously mentioned. A revision score of 2 is assigned when students repeat the same evidence already provided or a score of 1 is given if they fail to align their changes with the Feedback Level 1 messages; for example, instead of introducing new evidence they only provide explanations for the evidence they had used in their first draft. Feedback Level 2 focuses on the specificity and the elaboration of existing evidence; thus, a revision score of 3 is assigned if students add significant detail or explanation to more than one piece of evidence. Conversely, a score of 2 is assigned if students merely paraphrase the existing evidence, and a score of 1 is applied if students, contrary to the focus of the feedback message, add new evidence instead of elaborating on their existing evidence. Feedback Level 3 emphasizes explaining existing evidence and its connection to a claim. A revision score of 3 is assigned if students provide a strengthened explanation for more than one piece of evidence. A less successful revision may result from offering relatively brief or repetitive explanations (score 2), or from misalignment with the feedback message (score 1). This would be shown, for example, by students merely elaborating on their evidence without effectively connecting it to their claim. These detailed patterns associated with each score thus provide a nuanced guide for humans evaluating revisions.

To evaluate the quality of revisions, human raters began by identifying the changes students made to their essays. Each pair of essays, consisting of the initial and revised versions, was placed in separate Word documents. By using the "Compare Documents" feature in Word, the document highlighted areas where students added, deleted, or modified text. Then, taking into account the feedback level of the initial draft, human raters used the revision rubric (Appendix B) to determine the revision score.

Three human raters engaged in the evaluation process, which was divided into two phases. In the first phase, the primary rater, who played a crucial role in developing the rubric, trained the second rater to score the first 300 essay pairs. Sixty essay pairs were randomly selected from the three feedback levels and were coded by both raters. The interrater agreement for these pairs was 82% for exact matches and a Quadratic Weighted Kappa (QWK) of 0.74, demonstrating substantial consistency. In the second phase, the second rater, now experienced, trained the third rater to assess the remaining 300 essay pairs. This time, 30 essay pairs selected from the three feedback levels were double-coded for calibration. The interrater agreement reached 83% for exact matches and a QWK of 0.75, which again indicated a substantial level of reliability. The distribution of human revision scores at each feedback level is shown in Table 1.

	Revision Score 0 N (%)	**Revision Score 1** N (%)	**Revision Score 2** N (%)	**Revision Score 3** N (%)
Feedback Level 1	36 (26.67%)	40 (29.63%)	42 (31.11%)	17 (12.59%)
Feedback Level 2	53 (17.15%)	119 (38.51%)	104 (33.66%)	33 (10.68%)
Feedback Level 3	29 (18.59%)	56 (35.90%)	54 (34.62%)	17 (10.90%)

Table 1: Distribution of human revision scores at each feedback level

3 Experimental design

3.1 Experiment 1: Zero-shot prompt design (Baseline model)

In the initial experiment, we assessed GPT-4's capability in evaluating the quality of students' revisions to their text-based argumentative essays. The prompt was structured in the following order (see Appendix C for the prompt details):

1. Scoring task: This section outlined a clear scoring task for GPT-4. It introduced the stages where students were in their text-based argumentative writing tasks, having completed their first draft and then finished their second draft based on the feedback received. The feedback messages provided to students were incorporated into the prompt.

2. Writing task: This section introduced the text that formed the basis for the students' essays. The writing prompt was also included.

3. Detailed scoring rubric: The aforementioned revision rubric with the concrete revision patterns was included.

4. Student first and second drafts: To assess the quality of revisions, both the first and second drafts of student essays were provided.

3.2 Chain-of-Thought prompt design

We tested two different strategies of Chain-of-Thought (CoT) for improving the performance of GPT-4.

Experiment 2: One-shot CoT with human rater rationale

We provided GPT-4 with one example for each feedback level, all identified as successful revisions (holistic score of 3), accompanied by the human raters' rationale for their ratings (Appendix D). Considering that all essays came from fifth and sixth graders who were in the process of learning how to write argumentative essays, including successful revision examples in the prompt can aid GPT-4 in adjusting its scoring to reflect a more appropriate standard for young learners as opposed to the more advanced revisions that would be expected of adults. By presenting the rationale of human raters, our goal was to instruct GPT-4 to follow intermediate reasoning steps that human raters would apply. We further asked GPT-4 to provide a rationale for scoring before giving its score with the aim of eliciting a chain of reasoning.

Experiment 3: One-shot CoT with intermediate steps

To improve GPT-4's ability to use the rubric effectively, the rubric was transformed into a sequential flow of reasoning steps. This approach aimed to guide GPT-4 through the evaluation process in a step-by-step manner, closely simulating the decision-making pathway used by human raters (Figure 1). In addition, we also provided one example of successful revision for each feedback level in the prompt to support GPT-4 to adjust its scoring to reflect an appropriate evaluation standard for young students. We further asked GPT-4 to provide a rationale before giving its score with the aim of eliciting a chain of reasoning.

4 Results

4.1 Research question 1: How accurately can GPT-4 assess the revision quality of students' argumentative writing in comparison with human raters?

We conducted three experiments employing GPT-4 combined with CoT prompting strategies to assess their effectiveness in predicting the holistic scores for writing revision quality. Our primary evaluation metrics were Quadratic Weighted Kappa (QWK), which are widely used in automated essay scoring (AES) tasks.

	Zero-Shot	**One-Shot CoT** (Human rationales)	**One-Shot CoT** (Intermediate steps)
Exact Agreement	52.00%	54.50%	36.33%
Quadratic Weighted Kappa	0.60	0.60	0.46

Table 2: Overall revision score agreement rate

In the initial zero-shot prompting experiment, which served as our baseline, we observed an exact agreement rate of 52.00% and a QWK of 0.60, which suggested a moderate level of agreement between human raters and GPT-4 (Table 2). In our second experiment, we introduced a single example of a successful revision (revision score 3) along with the human rationale for that score at each feedback level. This approach improved the exact agreement rate to 54.50% while the QWK remained unchanged. Overall, by applying detailed rubrics with specific and concrete revision patterns corresponding to each score, GPT-4 demonstrated notable potential for assessing the quality of student revisions. However, while many studies indicate that including examples with human rating rationales greatly outperforms baseline models (e.g., Xia et al., 2024; Yancey et al., 2023), our second experiment only found a slight improvement in the exact agreement between human raters and GPT-4 when the one-shot CoT was applied.

Furthermore, the rubric used in the baseline and second experiment was developed from observations made by human raters adhering to the scoring procedure. As the rubric only contains the most common revision patterns under each revision score, the rubric may not capture the full depth of our evaluation criteria for student revision quality. Thus, we introduced a structured three-step scoring process as a novel form of Chain-of-Thought to assess whether GPT-4 could mimic the human thinking process during complex tasks. However, this approach yielded a significant decrease in agreement rates. Specifically, as shown in the third column in Table 2, the exact agreement rate decreased to 36.33%, while the QWK dropped to 0.46. The outcomes implied that a rubric with clearly defined patterns for student revisions outperforms the more explicit but abstract scoring process used by human raters.

4.2 Research question 2: How does GPT-4's performance in evaluating revisions vary across different levels of young students' argumentative writing abilities?

We further explored the extent to which the level of agreement between GPT-4 and human raters varied with students' argumentative writing skills. As previously described, we categorized students' initial drafts into three levels based on the number of pieces of evidence, specificity of evidence, concentration of evidence, and word count. Students with Level 1 drafts were advised to improve their writing by adding more evidence, while those with Level 2 and 3 drafts were guided towards more advanced revisions centered on the elaboration and explanation of the evidence provided. From Table 3, it's evident that GPT-4 exhibits a markedly higher level of agreement with human scoring when assessing revisions in Level 1 essays, a pattern that persists across all three prompting strategies. Especially when one-shot CoT prompting is applied, we observed a notable enhancement in the precision of scoring predictions for Level 1 essays in contrast to Level 2 and Level 3, with the exact agreement

	Zero-Shot			**One-Shot CoT** (human rationales)			**One-Shot CoT** (intermediate steps)		
	Level 1	Level 2	Level 3	Level 1	Level 2	Level 3	Level 1	Level 2	Level 3
Exact Agreement	60.00%	47.90%	53.21%	65.93%	50.16%	53.21%	55.56%	30.74%	30.77%
Quadratic Weighted Kappa	0.73	0.53	0.58	0.77	0.54	0.54	0.71	0.38	0.43

Table 3: Revision score agreement rate at each feedback level

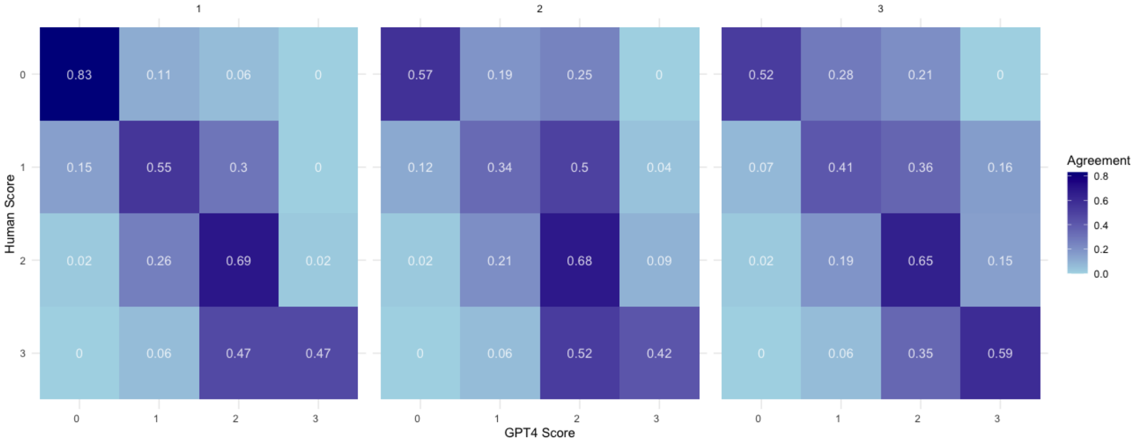


Figure 2: Confusion matrices of one-shot CoT prompting at each feedback level

increase from 60.00% to 65.93%, and the QWK from 0.73 to 0.77. This result suggests that GPT-4 is more likely to accurately evaluate the more concrete and straightforward task of adding evidence compared to evaluating evidence elaboration and explanation.

In contrast, the revision score agreement for Level 2 is lower than for Levels 1 and 3 across all three prompting strategies. Students with Level 1 or Level 3 essays were guided to focus exclusively on one aspect of revision: adding new evidence or adding explanations. Students with Level 2 drafts were in a middle position, as they were instructed not only to elaborate on the evidence but also to offer some explanations. When it comes to assessing the revision quality of draft 2, GPT-4 needs to examine revisions from two aspects, and this complexity may result in its inaccuracy. This result reemphasizes the potential limitations of GPT-4's accuracy in evaluating multifaceted tasks than simpler ones.

As the second experiment that applied one-shot CoT prompting demonstrated a relatively higher agreement among all three strategies, we focused on this condition for error analysis. Confusion matrices in Figure 2 reveal a strong consensus among humans and GPT-4 on the assignment of score 0 across all three levels, indicating no attempt at revision in the students' first drafts. Although the prediction of score 0 is highly accurate at Level 1, at Levels 2 and 3, despite being moderate, the accuracy of predicting score 0 diminished as GPT-4 tended to assign higher scores. A key factor could be that human raters might take into account the length of the student's initial draft when judging the revision effort, a nuance that GPT-4 might not effectively adjust for based on the student's writing proficiency.

Another noticeable trend is that GPT-4 tended to assign lower scores when human raters assigned a score of 3, consistent across all three levels (Figure 2). This discrepancy could stem from GPT-4's higher criteria for defining "a substantive improvement" in revisions. Table 4 provides an example from a Level 2 essay where the human rater assigned a score of 3 and GPT-4 assigned a score of 2. In the second draft, the student first improved one piece of evidence by adding a more relevant explanation of how providing money can contribute to better health. They also introduced new evidence regarding pollution issues, along with an explanation of how this supports their argument. Despite the repetition of ideas and less clear reasoning, the effort demonstrates a significant attempt at revision, as well as improvement in elaboration and explanation of existing evidence, from the perspective

First Draft	**Second Draft** (Student’s additions to their essay are indicated in red font)
We readers should fund money to space exploratons . . . one reason for the readers wanting to give money to the people is so that they can have food and shelter for their family.according to the text it states(1)” nearly half of all americans also have difficulty paying for housing,food,and medicine at some point in their lives.” ” in other countries, people are dying because they do not have access to clean water,medical care,or so simple solutions that prevent the spread of diseases.” (3)” for example,malaria, a disease spread by mosquito bites,kills many people in africa every year.” this quotation shows how we readers should donate money to the people who are living an unhappy life.this quotation makes it clear that we readers feel that the people who are homeless feels more important than the space exploration because they are poor. another reason is that people should get	We readers should fund money to space exploratons. . . one reason for the readers wanting to give money to the space exploration is because they want the people to be healthier and have a better and successful life to raise their children.According to pharagraph 2,it states” nearly half of all americans also have difficulty paying for housing, (2)” people are dying because they have no food or clean water to drink, also,it states” people needed medical instruments to keep the diseases from spreading and learn and develop body’s reaction area’s.” this quotation shows how my evidence makes it seem important that you should give money to people who are homeless and need to learn about medica instrument so they can clear their diseases and sickness.this quotation makes it clear that people would stay healthy by using medical instruments to cure their sickness and disease. another reason is that people need money so that they can clean and help earth stay healthy. according to pharagraph 3,it states ”(1) many scientist believe that pollution from burning fossils fuels is harming our air and oceans.” ” we need new,cleaner forms of energy to power cars,homes, and factories.” ” a program to develop clean energy could be viewed as a worthy investment.” this quotation shows how my evidence explains why space explorations also should still donate money to people so they can help earth get cleaned and to power factories and cars and also homes. this makes it clear that my evidence supports my reasoning state and also supports my claim.

Table 4: Example of student revision at feedback level 2

of a fifth or sixth-grader at least. In other words, humans appear more likely to consider students’ developmental level when scoring, a consideration that GPT-4 may overlook.

5 Discussion and conclusions

Revising is a very difficult skill to master, and many young students struggle to implement the feedback they receive (Roscoe et al., 2013). To foster the development of students’ revision skills, assessing revision quality and identifying revision patterns across various levels of writing proficiency is essential for providing targeted feedback to students on their revision efforts. With this aim, this study explored the potential of using a large language model, specifically GPT-4, to evaluate the quality of essay revisions aligned with the feedback messages students received from an AWE system.

First, our results suggest that GPT-4 has a great deal of potential for effectively evaluating writing revision quality. We used a detailed rubric providing specific revision patterns in the zero-shot (baseline experiment) prompting and one-shot CoT prompting and both approaches showed a moderate level of agreement between human raters and GPT-4. However, both CoT prompting strategies implemented in the study did not improve GPT-4 baseline performance. It is not altogether clear why this was the case as other researchers have found that CoT prompting tends to improve the accuracy of writing quality scores (Xia et al., 2024; Yancey et al., 2023). We note, however, that evaluating the quality of revisions in younger students’ essays may be a more complex task than assessing overall quality. It contains a series of evaluative steps beyond simply identifying revision patterns with a rubric. This includes interpreting feedback messages, identifying what was added in second drafts, and evaluating the alignment of those additions to the feedback. We recommend that future research explore additional prompting strategies to better address this complexity. For example, Tree-of-Thoughts prompting, which encourages LLMs to explore various ideas and assess intermediate steps in order to provide an optimal response (Yao et al., 2024), could be a useful way forward for generating more accurate assessments of complex writing processes.

Secondly, unlike studies that focus on adult writers such as college students, our research provides insight into the capabilities of LLMs to assess the writing produced by young students. We observed that GPT-4 tended to assign lower scores to revisions than human raters. One reason for this might be that fifth and sixth graders are still in the midst of developing their language as well as reasoning skills. The changes they make to their essays are constrained then by their overall ability to elaborate and explain their thinking in writing. Human raters took into account the age of students, and what they deemed reasonable to expect for revision at that age, and gave credit for effort (incremental changes) rather than only the quality of students' final product. Unlike human raters then, GPT-4 may lack knowledge of developmentally appropriate expectations for student writing which potentially affects its scoring accuracy. Therefore, LLMs would benefit from tailored training to adjust their criteria for "good" writing to be calibrated for different-aged students.

Limitations

Future research should consider the reliability of human ratings when evaluating GPT-4 scoring quality. While human raters remain the "gold standard" of writing evaluation, they are not always particularly consistent with one another (Brown, 2009; Cohen et al., 2018). In this study, we calculated only the overall reliability across three feedback levels among human raters, without specifically assessing the reliability at each feedback level. Further research is necessary to explore how human raters' scoring accuracy may vary across different levels of writing proficiency and within various scoring tasks, as well as how the reliability of human raters may influence the accuracy of automated scoring systems.

Moreover, this study focuses solely on exploring the potential of GPT-4, using it as an example among LLMs, for evaluating the quality of student revisions. Although GPT-4 has demonstrated impressive capabilities in various writing assessment tasks, alternative large language models, such as those outside the GPT family, may yield different results. Future research should investigate other LLMs, which would offer a more comprehensive understanding of the effectiveness of LLMs in assessing writing revisions.

Ethics statement

In our research, we assert that the dataset applied poses minimal risk regarding potential harm to individuals. The collection of writing essays from fifth and sixth-grade ELA teachers in Louisiana received approval from the Institutional Review Board at our institution. The intentions, procedures, and methods for collecting and using student essays are thoroughly detailed in the required consent forms provided to our teacher participants. Additionally, all collaborators adhere to stringent data privacy policies, ensuring further protection of participant information. Furthermore, our team is dedicated to enhancing participants' awareness and understanding of AWE systems. We meticulously developed interview questions aimed at uncovering their perceptions and concerns regarding the use of AWE technology. We also respect and value their suggestions on improving the design of the AWE systems to ensure they are intuitive, easy, and safe to use.

Acknowledgements

The research was supported by the National Science Foundation Award #2202347. The opinions expressed are those of the authors and do not represent the views of the foundation.

References

Mary Brindle, Steve Graham, Karen R Harris, and Michael Hebert. 2016. Third and fourth grade teacher's classroom practices in writing: A national survey. *Reading and Writing*, 29:929–954.

Gavin Thomas Lumsden Brown. 2009. The reliability of essay scores: The necessity of rubrics and moderation. *Tertiary assessment and higher education student outcomes: Policy, practice and research*, pages 40–48.

Yoav Cohen, Effi Levi, and Anat Ben-Simon. 2018. Validating human and automated scoring of essays against "true" scores. *Applied Measurement in Education*, 31(3):241–250.

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.

Richard Correnti, Lindsay Clare Matsumura, Laura S Hamilton, and Elaine Wang. 2012. Combining multiple measures of students' opportunities to develop analytic, text-based writing skills. *Educational Assessment*, 17(2-3):132–161.

Richard Correnti, Lindsay Clare Matsumura, Elaine Wang, Diane Litman, Zahra Rahimi, and Zahid Kisa. 2020. Automated scoring of students' use of text evidence in writing. *Reading Research Quarterly*, 55(3):493–520.

Richard Correnti, Lindsay Clare Matsumura, Elaine Lin Wang, Diane Litman, and Haoran Zhang. 2022. Building a validity argument for an automated writing evaluation system (erevise) as a formative assessment. *Grantee Submission*, 3.

Peter W Foltz and Mark Rosenstein. 2017. Data mining large-scale formative writing. *Handbook of learning analytics*, 199.

Steve Graham, Andrea Capizzi, Karen R Harris, Michael Hebert, and Paul Morphy. 2014. Teaching writing to middle school students: A national survey. *Reading and Writing*, 27:1015–1042.

Steve Graham, Michael Hebert, and Karen R Harris. 2015. Formative assessment and writing: A meta-analysis. *The elementary school journal*, 115(4):523–547.

Steve Graham and Dolores Perin. 2007. Writing next-effective strategies to improve writing of adolescents in middle and high schools.

Steve Graham and Karin Sandmel. 2011. The process writing approach: A meta-analysis. *The Journal of Educational Research*, 104(6):396–407.

Douglas Grimes and Mark Warschauer. 2010. Utility in a fallible tool: A multi-site case study of automated writing evaluation. *The Journal of Technology, Learning and Assessment*, 8(6).

Lindsay Clare Matsumura, G Genevieve Patthey-Chavez, Rosa Valdés, and Helen Garnier. 2002. Teacher feedback, writing assignment quality, and third-grade students' revision in lower-and higher-achieving urban schools. *The Elementary School Journal*, 103(1):3–25.

Lindsay Clare Matsumura, Elaine Lin Wang, Richard Correnti, and Diane Litman. 2023. Tasks and feedback: An exploration of students' opportunity to develop adaptive expertise for analytic text-based writing. *Assessing Writing*, 55:100689.

Elijah Mayfield and Stephanie Butler. 2018. Districtwide implementations outperform isolated use of automated feedback in high school writing. In *International Conference of the Learning Sciences*, volume 2128.

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.

Ben Naismith, Phoebe Mulcaire, and Jill Burstein. 2023. Automated evaluation of written discourse coherence using gpt-4. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 394–403.

NCES. 2012. The nation's report card: Writing 2011.

Corey Palermo and Margareta Maria Thomson. 2018. Teacher implementation of self-regulated strategy development with an automated writing evaluation system: Effects on the argumentative writing performance of middle school students. *Contemporary Educational Psychology*, 54:255–270.

Rod D Roscoe, Erica L Snow, and Danielle S McNamara. 2013. Feedback and revising in an intelligent tutoring system for writing strategies. In *Artificial Intelligence in Education: 16th International Conference, AIED 2013, Memphis, TN, USA, July 9-13, 2013. Proceedings 16*, pages 259–268. Springer.

Tamara P Tate, Jacob Steissa, Drew Baileya, Steve Grahamb, Daniel Ritchiea, Waverly Tsenga, Youngsun Moona, and Mark Warschauera. 2023. Can ai provide useful holistic essay scoring? *OSF Preprints*.

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.

Wei Xia, Shaoguang Mao, and Chanjing Zheng. 2024. Empirical study of large language models as automated essay scoring tools in english composition_taking toefl independent writing task for example. *arXiv preprint arXiv:2401.03401*.

Changrong Xiao, Wenxing Ma, Sean Xin Xu, Kunpeng Zhang, Yufang Wang, and Qi Fu. 2024. From automation to augmentation: Large language models elevating essay scoring landscape. *arXiv preprint arXiv:2401.06431*.

Kevin P Yancey, Geoffrey Laflair, Anthony Verardi, and Jill Burstein. 2023. Rating short l2 essays on the cefr scale with gpt-4. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 576–584.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Haoran Zhang, Ahmed Magooda, Diane Litman, Richard Correnti, Elaine Wang, LC Matsmura, 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 Feedback focus corresponding to each feedback level

Feedback Level 1 (Completeness & Specificity):

- Use more evidence from the article (Completeness)
- Provide more details for each piece of evidence you use (Specificity)

Feedback Level 2 (Specificity & Explanation):

- Provide more details for each piece of evidence you use (Specificity)
- Explain the evidence (Explanation)

Feedback Level 3 (Explanation & Connection):

- Explain the evidence (Explanation)
- Explain how the evidence connects to the main idea and elaborate (Connection)

B Rubric for assessing revision quality aligned with feedback message

Essay Level	**0—No Attempt** No content revision attempted	**1—Attempted, Not Aligned** Content revision attempted but not aligned with feedback message	**2—Aligned, Not Improved** Content revision aligned with feedback message but no/slight improvement in evidence use	**3—Aligned, Improved** Content revision improved evidence use in line with feedback message
Level 1	• No edits at all • Revision focused solely on writing mechanics. • Only several words added or changed.	• Student added evidence that is not directly related to the argument or text • Student provided explanation for evidence provided • Student elaborated on explanation they already attempted to provide. • Student connected evidence to argument	• Student added one relevant piece of evidence • Student added general discussion (without a specific quote or paraphrase) that supports the argument and is generally based in the text • Student added direct quotes to support paraphrases that were already there.	• Student added at least two relevant piece of evidence that are on the correct side of the argument
Level 2	• No edits at all • Revision focused solely on writing mechanics • Only a short line or two changed without significant content added.	• Student added evidence or details that are not directly related to the argument or text • Student added evidence, but did not add specificity (more details to evidence already provided) without any explanation • Student added empty explanation (i.e., "I included this evidence because it supports my point") • Student added explanations that did not connect to the argument or that contradict the argument • Student made minimal content-based edits of any sort considering the length of the entire essay	• Student added small details (at least 2 small instances) • Student added brief explanations of evidence (at least 2 small instances) • Student paraphrased existing evidence	• Student added relevant and solid details of evidence or explanations to at least two existing evidence
Level 3	• No edits at all • Revision focused solely on writing mechanics • Only a short line or two changed without significant content added	• Student added evidence or details that are not directly related to the argument or the text • Student added evidence or added more details to evidence without any explanation • Student added empty explanation (i.e., "I included this evidence because it supports my point") • Student added explanations that do not connect to the argument or that contradict the argument • Student made minimal content-based edits of any sort considering the length of the entire essay	• Student recycled same explanation for each piece of evidence • Student paraphrased existing evidence • Student only added one strong explanation for only one piece of evidence • Student added a decent explanation only at the end of the essay, not after each piece of evidence • Student added personal commentary, not explanation of evidence that connects to argument	• Student strengthened explanation for at least two pieces of existing evidence • Student provided strong connection between evidence presented to the overall argument

C GPT-4 prompt

Scoring task. 5th and 6th graders are learning how to write and revise text-based argumentative essays, particularly focusing on the use of evidence from the text. After they submit their first drafts, each student's work is assessed and categorized into levels—Level 1, Level 2, or Level 3—reflecting the quality of their writing. Based on the level their drafts are assigned, students receive corresponding feedback for Level 1, Level 2, or Level 3, which helps guide their revisions.

Level 1 feedback message concentrates on "Using more evidence from the article" and "Providing more details for each piece of evidence you use". Level 2 feedback message concentrates on "Providing more details for each piece of evidence you use" and "Explain the evidence". Level 3 feedback message concentrated on "Explain the evidence" and "Explain how the evidence connects to the main idea and elaborate".

Your role is to score the quality of revision from the first draft to the second draft based on a rubric that will be provided to you. The rubric comprises four ratings (0,1,2,3), focusing on evaluating whether students' revisions align with the feedback provided and if there is an improvement in their essays.

Writing task. This is the text the student needs to read before writing: A Question to Consider: Is space exploration really desirable when so much needs to be done on Earth? This is a question that has been asked for several decades and requires serious consideration. The arguments against space exploration stem from a belief that the money spent could be used differently – to improve people's lives. In 1953, President Eisenhower captured this viewpoint. He opposed the space program, saying that each rocket fired was a theft from citizens that suffered from hunger and poverty. Indeed, over 46.2 million Americans (15%) live in poverty. Nearly half of all Americans also have difficulty paying for housing, food, and medicine at some point in their lives. In other countries, people are dying because they do not have access to clean water, medical care, or simple solutions that prevent the spread of diseases. For example, malaria, a disease spread by mosquito bites, kills many people in Africa every year. It is possible to lower the spread of this disease by hanging large nets over beds that protect people from being bitten as they sleep. These nets cost only $5; however, most people affected by malaria cannot afford these nets. It is not just people that need help. The Earth is suffering also. Many scientists believe that pollution from burning fossil fuels (gasoline and oil) is harming our air and oceans. We need new, cleaner forms of energy to power cars, homes, and factories. A program to develop clean energy could be viewed as a worthy investment. Maybe exploring space should not be a priority when there is so much that needs to be done on Earth. Right now, the government spends 19 billion dollars a year for space exploration. Some people think that this money should be spent instead to help heal the people and the Earth.

Tangible Benefits of Space Exploration: People in favor of space exploration argue that 19 billion dollars is not too much. It is only 1.2% of the total national budget. Compare this to the 670 billion dollars the US spends for national defense (26.3% of the national budget), or the 70 billion dollars spent on education (4.8% of the budget), or the 6.3 billion dollars spent on renewable (clean) energy. The investment in space exploration is especially worthwhile because it has led to many tangible benefits, for example, in the area of medicine. Before NASA allowed astronauts to go on missions, scientists had to find ways to monitor their health under stressful conditions. This was to ensure the safety of the astronauts under harsh conditions, like those they would experience on launch and return. In doing this, medical instruments were developed and doctors learned about the human body's reaction to stress. In rising to meet the challenges of space exploration, NASA scientists have developed other innovations that have improved our lives. These include better exercise machines, better airplanes, and better weather forecasting. All these resulted from technologies that NASA engineers developed to make space travel possible. Even the problems of hunger and poverty can be tackled by space exploration. Satellites that circle Earth can monitor lots of land at once. They can track and measure the condition of crops, soil, rainfall, drought, etc. People on Earth can use this information to improve the way we produce and distribute food. So, when we fund space exploration, we are also helping to solve some serious problems on Earth.

The Spirit of Exploration: Beyond providing us with inventions, space exploration is important for the challenge it provides and the motivation to bring out the best in ourselves. Space exploration helps us remain a creative society. It makes us strive for better technologies and more scientific knowledge. Often, we make progress in solving difficult problems by first setting challenging goals, which inspire innovative work. Finally, space exploration is important because it can motivate beneficial competition among nations. Imagine how much human suffering can be avoided if nations competed with planet-exploring spaceships instead of bomb-dropping airplanes. We saw an example of this in the 1960's. During what is called the Cold War, the United States and Russia competed to prove their greatness in a race to explore space. They each wanted to be the first to land a spacecraft on the moon and visit other planets. This was achieved. It also resulted in many of the technologies and advancements already mentioned. In addition, the 'space race' led to significant investment and progress in American education, especially in math and science. This shows that by looking outward into space, we have also improved life here on Earth.

Returning to the Question All this brings us back to the question: Should we explore space when there is so much that needs to be done on Earth? It is true that we have many serious problems to deal with on Earth, but space exploration is not at odds with solving human problems. In fact, it may even help find solutions. Space exploration will lead to long-term benefits to society that more than justify the immediate cost.

This is the writing prompt: Consider the reasons given in the article for why we should and should not fund space exploration. Did the author convince you that "space exploration is desirable when there is so much that needs to be done on earth"? Give reasons for your answer. Support your reasons with 3-4 pieces of evidence from the text.

Scoring rubric with intermediate steps. We developed two types of rubric. The detailed rubric with concrete revision patterns would be introduced in Appendix C. The scoring rubric with intermediate steps was presented here:

Feedback Level 1. Step 1: Please compare the first draft and second draft, did the student show an attempt at content-level revision, considering the length of the first draft? If answer is no attempt or minimal attempt (including no edits at all, or only few words, revision focused solely on writing mechanics), please output score 0. Step 2: If yes, did the student's revision align with the feedback message, considering the text content? If answer is no (including that student provided explanation or elaborate on evidence for evidence provided), please output score 1. Step 3: If yes, did the student's revision show substantive improvement in the use of evidence? If answer is no improvement or slight improvement (including that student added one relevant piece of evidence, or student added direct quotes to support paraphrases that were already there), please output score 2. If yes (substantive improvement is that student added at least two solid and relevant piece of evidence that are on the correct side of the argument), please output score 3.

Feedback Level 2. Step 1: Please compare the first draft and second draft, did the student show an attempt at content-level revision, considering the length of the first draft? If answer is no attempt or minimal attempt (including no edits at all, or revision focused solely on writing mechanics, or only a short line or two changed without significant content added), please output score 0. Step 2: If yes, did the student's revision align with the feedback message? If the answer is no (including that student added new evidence but did not add more details to evidence already provided, or student added empty explanation, or student added explanations that did not connect to the argument or that contradict the argument, or student added personal commentary or non-text-based evidence), please output score 1. Step 3: If yes, did the student's revision show substantive improvement in the use of evidence, ? If answer is no improvement or slight improvement (including student added at least two small details, or student added at least two brief explanations of existing evidence, or student paraphrased existing evidence), please output score 2. If yes (substantive improvement is that student added relevant and solid details of evidence or explanations to at least two existing evidence), please output score 3.

Feedback Level 3. Step 1: Please compare the first draft and second draft, did the student show an attempt at content-level revision, considering the length of the first draft? If answer is no attempt or minimal attempt (including no edits at all, or revision focused solely on writing mechanics, or only a short line or two changed without significant content added), please output score 0. Step 2: If yes, did the student's revision align with the feedback? If the answer is no (including that student added evidence or added more details to evidence without any explanation, or student added empty explanation, or student added personal commentary, not explanation of evidence), please output score 1. Step 3: If yes, did the student's revision show substantive improvement in the use of evidence? If answer is no improvement or slight improvement (including that student recycled same explanation for each piece of evidence, or student paraphrased more than 1 existing evidence, or student only added one strong explanation for one piece of evidence, or student added at least two brief explanations of existing evidence, or student added a decent explanation only at the end of the essay, not after each piece of evidence), please output score 2. If yes (substantive improvement is that student strengthened explanation for at least two pieces of existing evidence, or student provided at least two pieces of strong connection between evidence presented to the overall argument), please output score 3.

D Examples of score 3 with the human rater rationale at each feedback level

Feedback Level 1:

- First draft: I am convinced that space exploration is desirable because space exploration helps us remain a creative society.It makes us strive for better technologies and scientific knowledge. This shows that people need more on earth than space. Another example is that space exploration will lead to long term benefits to society that more than justify the immediate costs. This shows that space exploration is desirable .This is why I am convinced that space exploration is desirable when so much needs to be done on space and earth.

- Second draft: I am convinced that space exploration is desirable because space exploration helps us remain a creative society.It makes us strive for better technologies and scientific knowledge. This shows that people need more on earth than space. Another example is that space exploration will lead to long term benefits to society that more than justify the immediate costs. This shows that space exploration is desirable .This is why I am convinced that space exploration is desirable when so much needs to be done. Another reason why space exploration is desirable is how scientist use monitors to check astronauts health before they go on an mission. This is another reason why space exploration is desirable. My next reason is, in addition ,the race led to significant investment and progress in american education ,especially in math and science. this shows that by looking outward into space ,we also improved life here on earth. This is why I am convinced that space exploration is desirable.

- Human rationale for scoring: This is Level 1 feedback, requiring "Using more evidence from the article" or "Providing more details for each piece of evidence used." The student attempted a content-level revision. The student added "Another reasons . . . we also improve life here on earth.", which seems to be an effort to add three text-based evidence to support their argument. Thus, the revision aligns with the feedback message and also results in a substantive improvement of the essay's evidence use. Therefore, the revision score is 3.

Feedback Level 2:

- First draft: Space exploration is desirable when there is so much that needs to be done on the earth. The space exploration can help solve some of the worlds problems. serious problem accrue on earth but the space exploration can fix some of them. Hunger problems, soil,crops,rainfall,droughts etc, can be solved by space exploration like the satellites that are around earth that monitor lots of land for the way food is produced and distributed. The text states "people on Earth can use this information to improve the way we produce and distribute food." This shows that the production of food and the way its distributed is going to be better if the scientist do the space explo- ration. The text also states "In rising to meet the challenges of space exploration, NASA sci- entist have developed other innovations that have improved our lives." Space exploration is desirable when there is so much that needs to be done on the earth. Earth has problems on it but scientist can solve them with space exploration. So space exploration is desirable to solve the needs of earth.

- Second draft: Space exploration is desirable when there is so much that needs to be done on the earth. The space exploration can help solve some of the worlds problems. serious problem accrue on earth but the space exploration can fix some of them. Hunger problems, soil,crops,rainfall,droughts etc, can be solved by space exploration like the satellites that are around earth that monitor lots of land for the way food is produced and distributed. The text states "people on Earth can use this information to improve the way we produce and distribute food." This shows that the production of food and the way its distributed is going to be better if the scientist do the space exploration.The way we distribute our food is important we have to make sure we have the right amount for everyone.The text also states "In rising to meet the challenges of space exploration, NASA scientist have developed other innovations that have improved our lives." This piece of evidence explains the way we face challenges on Earth,but that we can improve our lives a little better with the space exploration. Space exploration is desirable when there is so much that needs to be done on the earth. Earth has problems on it but scientist can solve them with space exploration. So space exploration is desirable to solve the needs of earth.The text states"Beyond providing us with inventions, space exploration is important for challenges it provides and the motivation to bring out the best in ourselves. Space exploration helps us remain a creative society." This shows that the space exploration helps in more ways than we thought, like we stay creative and it brings out our best side. Space exploration is desirable when there is so much that needs to be done on the earth. This shows how much we need space exploration.

- Human rationale for scoring: This is Level 2 feedback, requiring “Providing more details for each piece of evidence you use” or “Explain the evidence”. The student attempted a content-level revision. The student first added, "This piece of evidence explains the ...," which appears to be an attempt to provide an explanation for existing evidence. Additionally, the student added "the text states ’Beyond providing us ...’," which seems to be an effort to introduce detailed evidence along with an explanation for the argument. Thus, the revision aligns with the feedback message and also results in a substantive improvement of the essay’s evidence use. Therefore, the revision score is 3.

Feedback Level 3:

- First draft: They should get paid because 19 billion dollars a year for exploration. Most people think that this money should be spent instead of heal the people and the earth. Then 70 billion dollars spent on education (4.8% of the budget), or the 6.3 billion dollars spent on renewable (clean) energy. Before NASA allowed astronauts to go on the missions, scientists had to figure out how to monitor there health under any stress- ful conditions. They did this for the safety of the astronauts. NASA scientists have de- veloped other innovations that have improved our lives. NASA engineers developed to make space travel so they can do there mission. it is not just the people that need help. The Earth is suffering also. Many scientists believe that pollution from burning fossil fuels (Gasoline and oil) is harming our air and oceans. We need new, cleaner forms of energy to power cars, homes, and factories.

- Second draft: They should get paid because 19 billion dollars a year for exploration. Most people think that this money should be spent instead of to heal the people and the earth. Then 70 billion dollars spent on education (4.8% of the budget, or the 6.3 billion dollars spent on renewable (clean) energy. Before NASA allowed astronauts to go on the missions, scientists had to figure out how to monitor there health under any stressful conditions. They did this for the safety of the astronauts. NASA scientists have developed other innovations that have improved our lives. NASA engineers developed to make space travel so they can do there mission. it is not just the people that need help. the Earth is suffering also so that means that they need money to have the stuff to look and see what is going to happen in the future and there is a machine in space to see what the weather is going to be so they need money for that. It is important because like what is there is a tornado unexpected so they will not know how cold or what is going to happen there might be snow coming and we do not know. Many scientists believe that pollution from burning fossil fuels (Gasoline and oil) is haring our air and oceans. We need new, cleaner forms of energy to power cars, homes, and factories. They also need money to have satellite see if we did not have a satellite we would not know when a tornado would come so that means we would not be prepared for a tornado we would not be able to evacuate or not get water food for a flood we would know have anything if we were not prepared it would come unexpected that is why they need money for all the things like satellite so we can be prepared for any storm. I think we should keep giving them money because they are keeping us safe by making a satellite and telling us on the news so we can get the info so we should keep giving they money so we can stay safe the money is a reward for keeping us safe so they should get money.

- Human rationale for scoring: This is level 3 feedback, requiring “Explain the evidence” or “Explain how the evidence connects to the main idea and elaborate”. The student attempted a content-level revision. The student first added, "so that means that they need ..." which appears to be an attempt to provide an explanation for why innovation can improve life on the the earth, such as weather. Additionally, the student added "they also need moeny to have satellite..." which seems to be an effort to introduce detailed evidence of satellite along with an explanation for how satellite can prepare for storm. Thus, the revision aligns with the feedback message and also results in a substantive improvement of the essay’s evidence use. Therefore, the revision score is 3.

Automatic Crossword Clues Extraction for Language Learning

Santiago Berruti **Arturo Collazo** **Diego Sellanes**

Aiala Rosá **Luis Chiruzzo**

Instituto de Computación, Facultad de Ingeniería, Universidad de la República
Montevideo, Uruguay

Abstract

Crosswords are a powerful tool that could be used in educational contexts, but they are not that easy to build. In this work, we present experiments on automatically extracting clues from simple texts that could be used to create crosswords, with the aim of using them in the context of teaching English at the beginner level. We present a series of heuristic patterns based on NLP tools for extracting clues, and use them to create a set of 2209 clues from a collection of 400 simple texts. Human annotators labeled the clues, and this dataset is used to evaluate the performance of our heuristics, and also to create a classifier that predicts if an extracted clue is correct. Our best classifier achieves an accuracy of 84%.

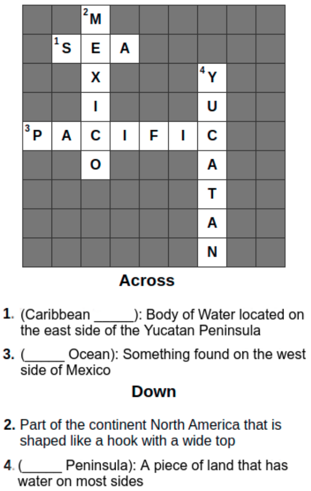
1 Introduction

This paper presents a series of experiments on automatically extracting clues from a text, that could be used to generate a crossword puzzle. Crosswords are a very interesting tool that can be used in educational contexts, in particular for developing vocabulary (Orawiwatnakul, 2013). In this work, we will focus on extracting words and generating definitions for crosswords in the context of teaching English as a foreign language, in particular for students at the beginner level.

A crossword (see Fig. 1) is a type of puzzle where words are arranged horizontally or vertically, and often intersect each other. The puzzle is presented with blank spaces where the letters should be, and is accompanied by the set of definitions of the target words. In our case, these words and definitions will be related to a text, for example an article or story that an English teacher wants to work with in class. The crossword in the figure could be obtained by processing the following article[1]:

(1) *Mexico is part of the continent North America. Mexico is shaped like a hook with a wide top. (...) On its west side is the Pacific Ocean. (...) A peninsula is a piece of land that has water on most sides. The Yucatan Peninsula has the Gulf of Mexico on its west and north sides. It has the Caribbean Sea on its east side.*

In order to do this, we must detect a set of interesting words from the text, extract their corresponding definitions, and create the crossword puzzle. In this work, we are not focusing on building the actual puzzle grid, but on extracting appropriate clues from the text that could be used to populate the crossword.



Across

1. (Caribbean _____): Body of Water located on the east side of the Yucatan Peninsula
3. (_____ Ocean): Something found on the west side of Mexico

Down

2. Part of the continent North America that is shaped like a hook with a wide top
4. (_____ Peninsula): A piece of land that has water on most sides

Figure 1: Possible crossword with clues extracted from example 1.

This kind of crosswords could be used as reading comprehension exercises, so it is expected that a student reads the text first, and then tries to solve the associated crossword. Notice that in this situation, the types of definitions we are trying to extract will generally be tied to the accompanying text, and would not exactly be dictionary definitions.

Throughout the text we will use the term

[1] Abridged version of the article "Where Is Mexico?" from ReadWorks.

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 381–390
June 20, 2024 ©2024 Association for Computational Linguistics

"definiendum" for a word that could appear in a crossword, and *"definition"* for a short phrase that defines that word. Likewise, when we mention a *"clue"*, we are referring to a <definiendum, definition> pair in this context.

In this project, we created a series of heuristics for extracting clues from simple texts, stories and articles. We used the heuristics to create a small annotated dataset of <definiendum, definition> pairs, labeled according to how correct they are to be used in a crossword and how grammatical they are. With this dataset, we trained several machine learning systems that try to predict if new clues would be suitable for creating a crossword.

The main contributions of this work are the following: 1) We present a set of heuristics that can extract clues from simple English texts. The heuristics range from simple linguistic patterns extraction to more complex question-answer generation, and could also combine information from different sentences in a text. 2) We annotated a dataset of 2209 clues, generated using our heuristics, with information about grammaticality and correctness as a clue for a crossword (i.e. it would be suitable to include this definition for this definiendum in the context of a crossword)[2]. 3) We did experiments on automatic classification of clues, with the best classifier achieving 84% accuracy and 78% macro-F1 for detecting correct clues.

The rest of the paper is structured as follows: Section 2 presents some relevant related work, section 3 describes the corpus we used and the heuristics we created for extracting clues, section 4 shows the quality evaluation of the extracted clues and presents the classifier we built, and finally section 5 presents some conclusions and future work.

2 Related Work

The works on automatic generation of crossword clues from texts are scarce. We comment below those that are closer to our objectives.

In (Percovich et al., 2019), the authors present two approaches to the generation of crossword puzzles, with the aim of using them for teaching English as a second language at the beginner level. On the one hand, a set of definitions organized by classical categories (e.g., colors, food, animals) is created, from which crossword puzzles are automatically generated according to the selected category. The definitions are extracted from different sources: existing children's dictionaries were used, and new definitions are also generated by applying patterns on Simple Wikipedia texts and filtering those that do not correspond to the expected categories by applying heuristics based on word embeddings. On the other hand, crossword puzzles are generated from texts entered by teachers, from which pairs <definiendum, definition> are extracted automatically, as in our work. For this, some heuristics based on information from a dependency parser are applied, using the verb "to be" as a central element, and each clue is extracted from a single sentence.

In (Rigutini et al., 2012), a traditional approach based on linguistic analysis tools is presented. A pipeline is applied to generate crossword clues from texts obtained by web crawling. The system processes the texts by applying different analyzers in sequence: sentence splitter, POS-tagger, chunker, and specific rules for the identification of subject, object and predicate (verbal or nominal) of each sentence. Then a finite state automaton is applied to detect which sentences are definitions, and finally, to generate the crossword clue, the subject of the sentence is removed. The system was used to create Italian crosswords.

In (Esteche et al., 2017), a system for crossword generation from Spanish news texts is presented. They use tools for linguistic analysis –a POS-tagger, a constituency parser, and a clause segmenter– and from the information they provide they define recursive regular expressions to extract clues from the texts. The paper presents a wide variety of patterns, and includes a tool implemented in Prolog to generate different crossword grids. This last task of actually generating the crossword grid has been explored in the past (Meehan and Gray, 1997; Botea, 2007), and is not particularly relevant from an NLP perspective, although some of the ideas in (Esteche et al., 2017) such as using different priority levels for words when building the grid might be relevant in an educational context to make sure the words that the teacher wants to highlight are included in the puzzle.

In (Katinskaia et al., 2018), a platform for language learning is presented. The platform includes crossword-based exercises created from stories. The crosswords are composed of words taken from the story, the student has to guess the words in

[2] The corpus is not yet available because we are waiting for the authorization of the text owners to publish it, we hope this will happen soon.

their correct grammatical form.

Some of our extraction heuristics that use linguistic patterns bear some resemblance to the classic method proposed by Hearst (1992) in the context of hyponyms extraction, although in that work there is an iterative step in which previously extracted information is used to generate new patterns from a large corpus. We have not tried the iterative process in this work, although a similar approach has already been explored in the context of clues extraction for crosswords in the past (Esteche et al., 2017), where it was unable to find new productive patterns.

A similar task to the one addressed in this paper is the generation of Question & Answering exercises for English teaching, aiming at the same objective, which is to evaluate the comprehension of a text. The extraction of question/answer pairs from texts can be used as input to generate clues for crossword puzzles, by means of some transformations, as we show below. In (Yao et al., 2022; Berger et al., 2022) neural approaches for generating Q&A exercises for teaching are presented, in (Morón et al., 2021) a similar work is presented using patterns based on different linguistic analyses (POS-tagging, semantic role labeling, coreference resolution, named entities recognition).

Another related NLP task is definition extraction, although with important differences from the problem addressed in this paper. Our goal is to extract clues for creating crosswords from texts. These clues may not make any sense outside the context of that text since they are not true definitions of the terms, in the strict sense of a dictionary definition. An important reference on definition extraction is SemEval-2020 shared task 6, "Definition extraction from free text with the DEFT corpus" (Spala et al., 2020), in which a specific corpus for definition extraction was used to train models. Fifty-one teams participated in this competition and most of them based their approaches on the use of pretrained language models.

3 Dataset and Clue Extraction

We created a number of heuristic rules or patterns that can be used to extract <definiendum, definition> pairs from simple texts in English. These rules were created by experimenting with a corpus of short texts, manually exploring and analyzing the frequencies of different expressions and patterns.

We used a dataset comprised of 400 short texts obtained from the ReadWorks website[3], an educational technology nonprofit organization. ReadWorks contains thousands of short texts and stories ranked in levels K, 1, 2, 3, 4, and 5 and categorized in the Lexile scale. The texts are written by experts and curated by educators, and could be non-fiction, fiction or poetry, within these three thematic areas: science, social studies and art. In our experiments we used 400 texts, most of them belong to level 1, and a few to level 2. These texts include short articles about history, geography and science, and some short stories.

Our clue detection and extraction rules begin with a pre-processing phase in which we perform coreference resolution using the AllenNLP tool (Gardner et al., 2018) and simple sentence splitting. Then we have a series of modules that apply different extraction patterns based on: syntax, Semantic Role Labeling (SRL), extended patterns that combine sentences, Named Entity Recognition (NER), and Question-Answering (QA). All these patterns extract rough clues, and we use a post-processing module to improve the shape of the definienda and definitions.

3.1 Syntax-based patterns

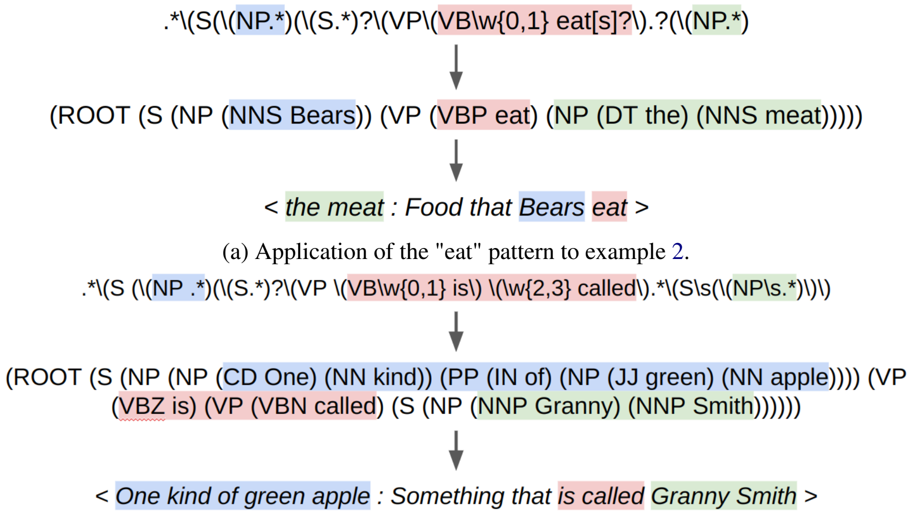
The first heuristic processes the constituency tree looking for some key verbs, and performs basic transformations to turn the phrase into a clue. We first analyzed our dataset searching for the most common verbs, trying to find ways in which the sentences these verbs took part in could be transformed into clues. Consider the following examples:

(2) *Bears eat the meat*

(3) *One kind of green apple is called Granny Smith.*

These examples use two frequent constructions in the corpus: the verb 'to eat' and the construction 'is called'. We crafted regular expressions for these frequent verbs that could be used to extract <*definiendum, definition*> pairs. These expressions operate over the text representation of the constituency tree, obtained using AllenNLP (Gardner et al., 2018), and use capture groups to define the parts of the text we want to extract. The patterns created for these verbs are shown in Fig. 2.

[3] https://www.readworks.org/


.*\(S(\(NP.*)(\(S.*)?\(VP\(VB\w{0,1} eat[s]?\).?(\(NP.*)
(ROOT (S (NP (NNS Bears)) (VP (VBP eat) (NP (DT the) (NNS meat)))))
< the meat : Food that Bears eat >
.*\(S (\(NP .*)(\(S.*)?\(VP \(VB\w{0,1} is\) \(\w{2,3} called\).*\(S\s(\(NP\s.*)\)\)
(ROOT (S (NP (NP (CD One) (NN kind)) (PP (IN of) (NP (JJ green) (NN apple)))) (VP (VBZ is) (VP (VBN called) (S (NP (NNP Granny) (NNP Smith))))))
< One kind of green apple : Something that is called Granny Smith >

(a) Application of the "eat" pattern to example 2.

(b) Application of the "is called" pattern to example 3.

Figure 2: Examples of application of syntactic patterns: a carefully tailored regular expression is applied to the textual representation of the constituency tree, and the capture groups are used to build the clue.

Note that it is actually not possible to use regular expressions to capture any type of expression from a constituency tree, but the type of simple sentences existing in the corpus, with low nesting levels, could mostly be treated with this tool.

As shown in Fig. 2, the types of clues extracted in this way could be rough around the edges, but the post-processing phase intends to fix some of these imperfections.

3.2 SRL-based patterns

Semantic role labels are a way of categorizing parts of a sentence as arguments of a predicate and the role they play in the described action (Palmer et al., 2005). The use of semantic role labels provides a more expressive way to define patterns that could capture some subtleties that regular expressions over constituency trees cannot. Semantic roles are, in a way, invariant to the syntactic position in the sentence, e.g. an argument with the role 'agent' could be acting as a subject or an object but still have the same semantic role.

We used the AllenNLP SRL analyzer and defined patterns that could be applied to these structures. In the SRL patterns, we look for combinations of phrases with role agent or theme (ARG0/ARG1) and phrases with role theme or attribute (ARG1/ARG2) associated to the same predicate. Several patterns were composed in this way, that work over the verbs like "to be", "to have", and "to like".

In the "to like" pattern, the analyzer already disambiguates the uses of "like" as a verb (predicate) or as a preposition, so the following examples are correctly resolved:

(4) *Bobby **likes** to play basketball.*

(5) *Bobby plays sports **like** basketball.*

The pattern for the verb "to live" is slightly different, because instead of an ARG1 it generally defined an ARGM which can either be a temporal or a location modifier. See the following examples:

(6) *Aztecs lived in Mexico.*

(7) *Dinosaurs lived in prehistoric ages.*

In example 6 the argument is labeled as `ARG-LOC`, so the clue is extracted as <*Mexico : Place where Aztecs lived*>, while in example 7 it is labeled as `ARG-TMP` and the clue is <*Prehistoric ages : Time when dinosaurs lived*>.

Besides looking for particular verbs, we built a more generic SRL pattern that captures any verb given that some valid combination of arguments is found. Optionally this pattern can also take some other types of modifiers, like in the following example:

(8) *A grown-up kangaroo can be bigger than a person.*

From example 8, the generic pattern can extract the clue <*kangaroo : something that can be bigger than a person*>, that includes the modifier "can" which is labeled as `ARGM-MOD` by the SRL module.

3.3 Extended patterns

The patterns seen so far work within the boundaries of one sentence, but we can create richer definitions if we combine the contents of more sentences. Consider the following example:

(9) *Bears are apex predators. They eat small mammals, like foxes.*

Two separate patterns (for the verbs "be" and "eat") could be applied independently, and if coreferences have been properly resolved, they both would have the same definiendum "Bears". In this kind of situations, where there is a nominative pattern ("be" or "is called") and a pattern that describes an action (such as "eat" or "live"), we can combine the definitions of the two patterns to create a new clue: <*Bears : apex predators that eat small mammals, like foxes*>.

3.4 NER-based patterns

In the NER patterns, we use the spaCy NER module (Honnibal et al., 2020) to find the named entities of the text and their categories. One pattern that already could capture the use of named entities was the "live" pattern, but in this case it is generalized when a named entity with a particular category is found. Take a look at the following examples:

(10) *Many chili peppers are grown in Mexico.*

(11) *Lebron James plays basketball at NBA.*

We have two cases with named entities of different categories. We can extract both names as definiendums, and use the category to create a definition tailored to that named entity. Mexico is classified as GPE (geo-political entity), so its clue would be <*Mexico : Place where many chili peppers are grown*>. Lebron James is classified as PER (person), so its clue ends up being <*Lebron James : Organization where Lebron James plays basketball*>.

3.5 QA-based patterns

Another way of generating clues is by casting the problem as a question answering task. First we use the NER module to extract named entities from the text that could be candidates to be used as definienda, together with some features like the category and number. Using the story as context, we create a question related to the named entity, and use the HuggingFace QA module[4] to generate an answer.

For example, using as context a story about the Great Sphynx, the process could detect the terms "Egypt", "Ancient Egyptians" and "Africa" as candidates. The following are the questions the process creates for those candidates, and the answers given by HugginFace QA:

- *What is Egypt? || A country in Africa.*
- *What are Ancient Egyptians? || They made the statue by cutting into a huge rock.*
- *What is Africa? || Egypt is a country.*

The first answer is a definition that fits perfectly, creating the clue <*Egypt : A country in Africa*>, but the other two are not correct in this context. In order to improve the quality of the clues, we filter out answers with a low confidence score as predicted by HuggingFace QA.

3.6 Post-processing

As mentioned above, some of the clues extracted might not be directly usable in crosswords, but we have a post-processing phase that can transform some of them to make them better. Consider the following examples of clues extracted with the patterns:

1. <*A snake : a reptile that moves its tail*>
2. <*Beauty and charm : Features Cleopatra had*>
3. <*Solar system : the name for the sun, planets, and other smaller bodies*>
4. <*Statue of Liberty : a symbol of freedom*>

The first step of post-processing is using NLTK POS tagger (Bird, 2006). There are different transformations that could be done after identifying the POS tags. If the definiendum is a determiner and a noun as in the first case, we can just drop the determiner. When a coordination is found, as in the second case, we can forget the conjunction and create two different clues with the remaining words. In other cases we select one or more words from the definiendum, giving preference to nouns, and move the rest of the words to the definition (third case). In the fourth case, as both words are good definienda, we create two different clues, keeping the remaining words in the definition.

After this process, the transformed clues are the following:

[4] https://huggingface.co/tasks/question-answering

1. *<snake : a reptile that moves its tail>*
2. *<beauty : Feature Cleopatra had>*
3. *<charm : Feature Cleopatra had>*
4. *<system : (Solar ___) the name for the sun, planets, and other smaller bodies>*
5. *<Statue : (___ of Liberty) a symbol of freedom>*
6. *<Liberty : (Statue of ___) a symbol of freedom>*

Sometimes the patterns tend to return underspecified definitions, like "something that is called Granny Smith" or "something that can be bigger than a person". These definitions are not fit for a crossword as they describe the terms too vaguely. As described above, the NER based pattern uses the named entity category to specify this, indicating whether the referred term is a person, location, organization, etc., while the SRL pattern can sometimes infer a more specific term for location or temporal modifiers.

However, this information is not available in all cases, so we implemented a heuristic based on the WordNet ontology (Miller, 1995; Fellbaum, 1998) that tries to improve this. WordNet is a lexical database that contains thousands of terms taxonomically structured by the hypernymy/hyponymy relation. The idea is to replace the vague term for a category that is still hypernymy of the definiendum, but is simple enough for students at the beginner level. In our case, we considered a list of simple categories that are generally part of the beginner level curricula: *animal, food, fruit, clothing, city, country, region, location, instrument, plant, tool, activity, action, relative, feeling, sensation*.

The heuristic tries to visit all the hypernyms of a definiendum and stops when one that belongs to our simple category list is found, otherwise, if we reach the most abstract "entity" term and no suitable candidate was found, we keep the first term of the definition as "something".

4 Experiments

After running our heuristics methods on all 400 texts of our dataset, our process generated 2321 <definiendum, definition> pairs. However, the quality of these clues might be very variable, depending on the pattern and on the text they were extracted from. It is very important to analyze which clues were correct and could be used for crosswords, and also we are very interested in making the whole process more accurate. One way to do this would be having a classifier that could discriminate if a new clue was correct or not according to some criteria. In this section, we describe the annotation of our corpus and the classifier we built.

4.1 Annotation

First of all, we annotated manually all the generated clues. Eleven annotators participated in this process[5], and they were asked to answer two questions for each clue: First, if the clue could be considered correct in the context of a crossword, considering that the person solving the puzzle would have read the corresponding text. Secondly, if the clue is grammatically correct.

After an initial annotation round that was used to discuss criteria and labeling conventions, we noticed that there was a third dimension we wanted to address. There were cases in which the original text had mistakes (probably transcription errors) that made the extracted clues unusable, these cases were to be marked as invalid and would be left out of the final corpus.

Each annotator was given a spreadsheet with the following information:

- **Definiendum:** Word to guess in the crossword.
- **Definition:** Text that defines the definiendum.
- **Context:** Main sentence were the clue was extracted from.
- **Text Name:** Name of the original text, so it could be checked for further context.
- **Method:** Heuristic pattern used to generate the clue.

They would also have the full texts that were needed for understanding their clues. The annotators had to indicate if the clue was valid or invalid (due to errors in the text), if it was correct (for a crossword), and if it was grammatical.

4.2 Analysis

All the 2321 clues originally extracted by the heuristics were considered for the annotation. In

[5] The annotators were researchers that took part in this work and a related project, and students that were compensated with undergraduate credits.

total, 112 of them were deemed as invalid because of errors in the texts, and were not considered for the rest of the analysis. The following is an example of a clue that is invalid, because the text contained a transcription error:

- **Context:** This painting Photos.comis titled Breaking Home Ties. It was painted by Thomas Hovenden, an Irish-born artist.
- **Definiendum:** Irish
- **Definition:** Breaking Home Ties

Some examples of clues that were considered correct and grammatical:

- **Context:** The White House has a swimming pool and a movie theater.
- **Definiendum:** Pool
- **Definition:** (Swimming ___) Something The White House has

- **Context:** So Franklin D. Roosevelt came up with plans to add more jobs.
- **Definiendum:** Roosevelt
- **Definition:** (Franklin D. ___) President that came with plans to add more jobs

The following is an example of a clue that could be considered correct, but is ungrammatical:

- **Context:** Green iguanas eat leaves, flowers, and fruit
- **Definiendum:** iguanas
- **Definition:** (Green ___) large that eat leaves, flowers, and fruit

The definition should be changed to something like “large animal that eats...” to be considered grammatical.

Table 1 shows the number of clues extracted by each pattern, and the corresponding values of correctness and grammaticality according to the annotators. The first thing we can notice is that some patterns are much more productive than others: all the SRL patterns were very productive, but especially the extended pattern that combines a sentence with the verb “to be” and another sentence generates a lot of clues, mainly because it could combine the already productive “to be” pattern with any other related clue. If we analyze the correctness and grammaticality of the clues, it is interesting to see that the SRL patterns once again are the most trustworthy: except the generic SRL pattern all the rest are very accurate in terms of grammaticality, and also mostly correct. On the other hand, the QA pattern was an under-performer, obtaining very few clues from the texts, and even then most of them were wrong, even if we set a confidence threshold for the generation. Exploring better and more powerful QA generation models (e.g. Yao et al. (2022); Berger et al. (2022)) would be necessary to improve this pattern.

Pattern	Total	Correct	Gramm.
`is called`	35	48%	86%
`eat`	53	83%	87%
`live`	15	13%	60%
`SRL have`	276	65%	90%
`SRL like`	54	65%	94%
`SRL live`	74	84%	88%
`SRL to be`	538	81%	96%
`SRL gen.`	217	75%	82%
`to be ext.`	887	70%	83%
`NER`	49	71%	86%
`QA`	11	28%	81%
Total	2209	72%	87%

Table 1: Number of clues extracted by each pattern from the whole dataset, together with their average correctness and grammaticality according to the manual annotation. The 112 invalid clues are not included in this table.

4.3 Classifier

Using this annotated set, we performed a series of experiments on creating a classifier that could automatically determine if a given clue is correct or not. For these experiments, we split the set in 80% for training and 20% test, and we used 5 fold cross validation for parameter tuning.

The different classification models we experimented with are the following:

Centroid distance baseline Based on the simple classifier presented in (Percovich et al., 2019), we obtain the FastText embeddings (Bojanowski et al., 2017) centroid of the context sentence, and of the definiendum and definition pair, and we calculate the Euclidian distance between them. Then we experimentally determined a distance threshold that maximized the F1 metric.

Machine learning methods Using a representation that takes the FastText embeddings of the context, definiendum and definition, we experimented with several classical machine learning models (Kowsari et al., 2019): KNN, Naïve Bayes, Decision Trees, Gradient Boosting and MLP.

Deep learning methods We carried out experiments with BERT based models, inspired by (Yao et al., 2022), which used a similar model for ranking automatically generated question-answer pairs. For these experiments we used the BERT (Devlin et al., 2018) and DistilBERT (Sanh et al., 2019) pretrained models, finetuned to our data with the HuggingFace Transformers (Wolf et al., 2019) `AutoModelForSequenceClassification` class. In both cases, the input is the context, the definiendum and the definition, separated by `SEP` tokens.

Table 2 shows the results of these experiments, and we can see that both BERT-based models outperform the rest, DistilBERT being the best model for our task.

Model	Accuracy	Macro F1
Centroid	0.66	0.58
KNN	0.72	0.57
Dec. Tree	0.64	0.55
Grad Boosting	0.71	0.59
MLP	0.73	0.59
NB	0.66	0.49
BERT	0.77	0.70
DistilBERT	**0.84**	**0.78**

Table 2: Accuracy and Macro F1 of the classifiers that predict the correctness of a clue.

5 Conclusions

We presented some experiments on automatic extraction of clues for crosswords from simple texts, considering a clue as a <definiendum, definition> pair that could be used in a crossword puzzle. We created several heuristic patterns for detecting and extracting clues using different NLP tools, like constituency parsing, SRL, NER and QA. With these heuristics, we extracted 2209 clues from a dataset of 400 documents from ReadWorks, and annotated them with information about correctness and grammaticality. The best heuristic patterns for extracting clues, according to our annotation, are the ones based on SRL.

Using our annotated dataset, we trained several classifiers on the problem of detecting whether an extracted clue is correct for a crossword. The best model for this turned out to be a DistilBERT model finetuned on our training data, obtaining 84% accuracy and 78% macro F1.

In the future, we want to explore the possibility of using large language models such as GPT or LLAMA for this task, which have shown promising results according to some preliminary experiments. We also want to explore the possibility of improving the QA based pattern by using better QA extraction modules. Currently we are in the process of testing our extraction system integrated to a crossword generation tool in a real case (Chiruzzo et al., 2022), with school children that are beginning to learn English, which would give us a better sense about how well our heuristics work and how they can be improved.

6 Ethics Statement

We understand that by using pretrained statistical NLP tools, our work could be infusing undesired biases in the results. This is especially dangerous in the situation we want to use the system, which is the context of a classroom with school children. Because of this, we consider that the results obtained by this tool must not be used directly by the students, but the supervision of a teacher is always necessary. In the system we are building, a teacher can automatically extract clues from a text and create a crossword, but they always have the possibility to inspect the generated clues in order to modify or remove any term or definition that might not be suitable, before the crossword is presented to students.

7 Limitations

In the experiments described in this paper, we have worked only with ReadWorks stories. These are texts designed to be simple and easy to read, and intend to be varied in terms of contents, but nonetheless they are only one data source and this means our process might end up be too tailored to the style and vocabulary of these texts and not generalize well to other sources.

Furthermore, given that these texts are very short, during our experiments we found that our heuristics generally can extract very few clues from each text. On average we can extract around four <definiendum, definition> pairs from a text (this can be noticed in the numbers presented in Table 1), which might be too few for creating an engaging crossword. We are working on improving the extraction process to generate more clues, but a combination with other methods such as including dictionary definitions of related words, especially short ones that could fill crossword gaps, would be advisable to build more complete and interesting crosswords.

Besides our heuristic patterns, we made some experiments to extract clues with more modern large language models (LLM) techniques which seemed promising. However, due to the limitations of our application servers, we decided to use more traditional methods because they are less demanding in terms of computational resources.

Acknowledgements

We want to thank ReadWorks for letting us use their simple English texts in the context of this research.

References

Gonzalo Berger, Tatiana Rischewski, Luis Chiruzzo, and Aiala Rosá. 2022. Generation of english question answer exercises from texts using transformers based models. In *2022 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, pages 1–5. IEEE.

Steven Bird. 2006. NLTK: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72.

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.

Adi Botea. 2007. Crossword grid composition with a hierarchical csp encoding. In *Proceeding of the 6th CP Workshop on Constraint Modelling and Reformulation, ModRef-07*.

Luis Chiruzzo, Laura Musto, Santiago Góngora, Brian Carpenter, Juan Filevich, and Aiala Rosá. 2022. Using nlp to support english teaching in rural schools. In *Proceedings of the Second Workshop on NLP for Positive Impact (NLP4PI)*, pages 113–121.

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*.

Jennifer Esteche, Romina Romero, Luis Chiruzzo, and Aiala Rosá. 2017. Automatic definition extraction and crossword generation from spanish news text. *CLEI Electronic Journal*, 20(2):6–1.

Christiane Fellbaum. 1998. *WordNet: An electronic lexical database*. MIT press.

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.

Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 volume 2: The 14th international conference on computational linguistics*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, Adriane Boyd, et al. 2020. spaCy: Industrial-strength natural language processing in python.

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).

Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. 2019. Text classification algorithms: A survey. *Information*, 10(4):150.

Gary Meehan and Peter Gray. 1997. Constructing crossword grids: Use of heuristics vs constraints. *Proceedings of Expert Systems*, 97:159–174.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Martín Morón, Joaquín Scocozza, Luis Chiruzzo, and Aiala Rosá. 2021. A tool for automatic question generation for teaching english to beginner students. In *2021 40th International Conference of the Chilean Computer Science Society (SCCC)*, pages 1–5. IEEE.

Wiwat Orawiwatnakul. 2013. Crossword puzzles as a learning tool for vocabulary development. *Electronic Journal of Research in Education Psychology*, 11(30):413–428.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

Analía Percovich, Alejandro Tosi, Luis Chiruzzo, and Aiala Rosá. 2019. Ludic applications for language teaching support using natural language processing. In *2019 38th International Conference of the Chilean Computer Science Society (SCCC)*, pages 1–7. IEEE.

Leonardo Rigutini, Michelangelo Diligenti, Marco Maggini, and Marco Gori. 2012. Automatic generation of crossword puzzles. *Int. J. Artif. Intell. Tools*, 21.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Sasha Spala, Nicholas Miller, Franck Dernoncourt, and Carl Dockhorn. 2020. SemEval-2020 task 6: Definition extraction from free text with the DEFT corpus. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 336–345, Barcelona (online). International Committee for Computational Linguistics.

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*.

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.

Anna Karenina Strikes Again: Pre-Trained LLM Embeddings May Favor High-Performing Learners

Abigail Gurin Schleifer[1] **Beata Beigman Klebanov**[2] **Moriah Ariely**[1] **Giora Alexandron**[1]
[1] Weizmann Institute of Science, Rehovot, Israel
[2] ETS, Princeton, USA
{abigail.gurin-schleifer,moriah.ariely,giora.alexandron}
@weizmann.ac.il
bbeigmanklebanov@ets.org

Abstract

Unsupervised clustering of student responses to open-ended questions into behavioral and cognitive profiles using pre-trained LLM embeddings is an emerging technique, but little is known about how well this captures pedagogically meaningful information. We investigate this in the context of student responses to open-ended questions in biology, which were previously analyzed and clustered by experts into theory-driven Knowledge Profiles (KPs). Comparing these KPs to ones discovered by purely data-driven clustering techniques, we report poor discoverability of most KPs, except for the ones including the correct answers. We trace this 'discoverability bias' to the representations of KPs in the pre-trained LLM embeddings space.

1 Introduction

Classifying students into behavioral or cognitive profiles using unsupervised cluster analysis techniques is a common application of machine learning to educational data (Le Quy et al., 2023; Martin et al., 2023; Ariely et al., 2024; Rastrollo-Guerrero et al., 2020; Bovo et al., 2013). Recently, there has been a growing interest in applying this methodology to textual student responses that are decoded using pre-trained large language models into vectorized embeddings in semantic spaces (Martin et al., 2023; Wulff et al., 2022; Masala et al., 2021). The operational appeal of this approach is that it minimizes the need for expert knowledge, which is costly to inject through human labeling procedures (Nehm and Haertig, 2012; Tansomboon et al., 2017; Li et al., 2023; Ariely et al., 2024). However, the validity of patterns discovered this way depends on the ability of the embeddings to maintain the pedagogically meaningful information that existed in the original, textual representations of responses (Devlin et al., 2018; Seker et al., 2022) and of the algorithmic method to discover them. Evaluation of emergent profiles is often done in terms of the internal quality of the clustering, as data is usually not available to estimate the extent to which the discovered profiles align with a pedagogically meaningful representation of the responses. Without such an evaluation, a loss of important information can be overlooked, potentially leading to sub-optimal educational decisions that rely on this analysis (Le Quy et al., 2023).

To investigate whether this hypothesized risk manifests in real-life educational context, we utilize student answers to two constructed response questions in high school biology. The data was previously analyzed by a team of biology education researchers and experienced teachers, and graded according to a theory-driven detailed analytic rubric that is based upon the Causal-Mechanical Explanation framework (Ariely et al., 2024; Salmon, 2006). The rubric contained 10 (item 1) or 11 (item 2) binary categories, each checking for the occurrence of a specific key piece of information in the response. Using these human-generated binary vectors of length 10 (11), the responses were clustered using a KMeans algorithm into a set of 6 (7) Knowledge Profiles (**KPs**) that were found by teachers to encapsulate specific patterns of errors.

The validity of the KPs was evaluated in several ways. First, human experts conducted a qualitative analysis to assess whether each KP captures a specific and distinct pattern of errors. Second, we analyzed the results computationally, showing that i) the KPs were consistent across the two items, namely, revealing the same type of conceptual errors; and ii) the *learners* tended to exhibit the same type of conceptual error (KP) in both items. Third, we conducted an in-class formative assessment intervention study that provided automated guidance to students based on their KP, and showed significant improvement in their performance on a different prompt that measures the same conceptual knowledge. These analyses provided strong evi-

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 391–402
June 20, 2024 ©2024 Association for Computational Linguistics

dence that the KPs capture pedagogically meaningful information (for full details, see Ariely et al. (2022, 2024)).

Using these data, we are in a position to answer two research questions:

RQ1 What is the correspondence between clusters that are computed from pre-trained LLM embeddings of student responses and theory-based KPs?

To preview the result, we find that two clustering techniques that are commonly used for such tasks (Le Quy et al., 2023) – KMeans (Lloyd, 1982) and HDBSCAN (McInnes et al., 2017) – largely fail to discover the KPs though retrieval is somewhat better for the profile containing the correct responses. Following up on this finding, we go 'upstream', to the pre-trained embeddings, and investigate:

RQ2 How well are the KPs represented in the pre-trained embeddings space?

Our results reveal a strong relationship between the quality of the responses in the profile (correct or various degrees of incorrect) and the shape and density of its embeddings-based representation. We refer to this relationship as an 'Anna Karenina principle' and tie it to the profile discovery failure we observed in RQ1.

The contribution of this work is twofold. First, it is the first to demonstrate the Anna Karenina principle in the context of pre-trained representation of student responses to open-ended questions. Second, our results suggest that, in some cases, out-of-the-box pre-trained LLM embeddings may be a pedagogically unsound basis for profile discovery.

2 Related Work

2.1 NLP-based profiling of constructed responses in science education

Open-ended items require students to develop and construct their answers, reflect on their knowledge, and integrate it with new ideas (Fellows, 1994). Reasoning and evidence-based defense of an argument is key for testing scientific hypotheses (Toulmin, 2003). Therefore, constructing causal explanations is an essential skill for students of science to learn (Ariely et al., 2024; Martin et al., 2023); practice and high-quality feedback are key elements in helping students master the skill (Hattie and Timperley, 2007; Gerard and Linn, 2016; Tansomboon et al., 2017).

Analyzing open-ended items to provide feedback is a time-consuming, complex task. Automating some of the analyses for assessment and feedback purposes is promising for supporting teaching and learning (Tansomboon et al., 2017; Gerard and Linn, 2016; Ariely et al., 2023).

Most systems for automated evaluation of scientific explanations to date had been designed in the supervised machine learning framework (Schleifer et al., 2023; Sung et al., 2019; Riordan et al., 2020; Kumar et al., 2019; Mizumoto et al., 2019; Li et al., 2021). Among the unsupervised approaches, Masala et al. (2021) extracted the main takeaways from students' feedback on different components in academic courses, using KMeans to cluster pre-trained BERT embeddings of students' feedback. Martin et al. (2023) applied HDBSCAN over pre-trained LLM embeddings and to find emergent argumentation patterns' characteristics. Wulff et al. (2022) investigated HDBSCAN clustering over LLM embeddings to evaluate the attention of preservice physics teachers to classroom events elicited from open-ended text responses. A semi-supervised coding method in which homogeneous clusters receive the same coding automatically and heterogeneous clusters are fully labeled by humans was proposed by Andersen et al. (2023) and applied to student responses to PISA items.

2.2 Biases in pre-trained LLMs

While LLMs are powerful meaning representations that undergird the state-of-art systems on a wide range of NLP tasks, they are also known to exhibit a plethora of social biases that could lead to social harm when the models are used in downstream tasks (Bender et al., 2021). In a recent review of the current state of research on LLM bias evaluation, Goldfarb-Tarrant et al. (2023) criticize the field for focusing heavily on the upstream, pre-trained LLMs, in most cases without considering the connection to a specific task the LLMs is being put to (68% of the reports reviewed), citing this as a threat to the predictive validity of bias measurements.

In fact, the literature that does consider the connection between upstream (intrinsic) and downstream (extrinsic) behavior suggests that it is not straightforward. Considering static embeddings (e.g., word2vec) and a commonly used bias test, the Word Embedding Association Test (WEAT) (Caliskan et al., 2017), Goldfarb-Tarrant and colleagues (Goldfarb-Tarrant et al., 2021) found no clear relationship with performance of models using the embeddings, as measured by differences in precision and recall of retrieval of the target construct on data from privileged and non-privileged groups. Extension to contextual embeddings and a wider range of tasks and measures yielded similar results (Cao et al., 2022; Kaneko et al., 2022). Our contribution extends the discussion towards social constructs beyond the typically considered demographic attributes such as gender, race, ethnicity, age towards a distinction that is particularly relevant when dealing with learner data – that of learners at the more or less advanced state of understanding of the phenomenon under consideration. We are not aware of prior work comparing LLM representations based on knowledge-related profiles; the closest finding in the literature are examples of poorer performance of LLM-based systems on data produced by English language learners with respect to native speakers of English (Baffour et al., 2023). Additionally, we explore LLMs in a relatively low-resource language (Hebrew) in contrast to the bulk of current work that focuses on English or other high-resource languages: In the 90 LLM bias studies evaluated by Goldfarb-Tarrant et al. (2023), only two report results in a language that is not highly resourced.

3 Data

The data consists of 669 student responses to two open-ended items in high-school biology, collected anonymously from students in grades 10-12 from about 25 high schools of varied demographics and socioeconomic status (based on location) across Israel. Gender distribution was 70% females (typical to the gender distribution among high-school biology majors in Israel). The items deal with the connection between respiration and energy in physical activity in the context of smoking (**Q1**) and anemia (**Q2**), taught as part of the core topic "The human body". The items were human-scored using a similar analytic rubric containing 10 (Q1) or 11 (Q2) categories (Ariely et al., 2024). All rubric categories are binary, each targeting specific information that needs to be mentioned in a correct response, such as "the role of hemoglobin in oxygen transportation" or "changes in cellular respiration rate". The resulting binary vectors were clustered using KMeans; the clusters were analyzed by experienced teachers and ranked from 1 to 6 (Q1) or 7 (Q2) with larger numbers corresponding to clusters with more severe errors. We denote these clusters *Knowledge Profiles*, and index them from 1 (KP1) to 7 (KP7). See Ariely et al. (2024) for a full description of the items and the assessment framework. The items and examples of student responses and their mapping into KPs can be found in Appendix 1.

For the purposes of the analysis presented in this paper, all responses were represented using rich contextualized vectors – embeddings produced by a pre-trained Large Language Model (LLM). The LLM being used, AlephBERT (Seker et al., 2022), is state-of-the-art for Hebrew. It was trained on a large corpus of the Hebrew language, including: Twitter tweets, Hebrew Wikipedia, and the Hebrew subset of the Oscar (Suárez et al., 2020) dataset. AlephBERT has the same architecture as BERT (Devlin et al., 2018): 12 layers, 110M parameters, and 12 attention heads. It was trained on a 52K-word Hebrew vocabulary on masked token prediction task, and on the Hebrew language tasks: word segmentation, part-of-speech tagging, and full morphological tagging. It was further trained on the tasks of sentiment analysis and named entity recognition.

4 Methods

To evaluate whether raw LLM embeddings carry useful knowledge for unsupervised profiling of responses, we experimented with two common clustering approaches (Le Quy et al., 2023), KMeans (Lloyd, 1982) and HDBSCAN (McInnes et al., 2017), which implement different clustering mechanisms. The first discovers convex-shaped clusters; its mechanism is centroid-based and applies an Euclidean distance function. The second is density-based and can be applied with various distance metrics, e.g., a metric induced by cosine-similarity, and the clusters may have various shapes. Both approaches were used previously for profile discovery in constructed response data (Ariely et al., 2024; Martin et al., 2023; Wulff et al., 2022). Experiments were conducted in Python, using scikit-learn (Pedregosa et al., 2011), SBERT (Reimers and Gurevych, 2019) and Pytorch (Paszke et al., 2019).

4.1 KMeans

The KMeans is a widely used algorithm (Lloyd, 1982). The algorithm is initiated with a specified number of clusters and a random initialization of their centroids. The clustering approach minimizes the within-cluster sum of squared distances, i.e., Euclidean distance. KMeans clusters are convex and all samples are assigned to a cluster, i.e., there are no outliers. Convexity means that for every two points in the cluster, a straight line between them also lies within the cluster.

4.2 Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN)

Another clustering approach, which is more promising in the context of LLMs' embeddings (Martin et al., 2023) is the HDBSCAN (McInnes et al., 2017) algorithm. The approach here is creating a mutual reachability graph where *core* samples are points in areas of high density. A cluster is a set of *core* samples and a set of *non-core* samples that are neighbors of *core* samples but are not *core* themselves. *Non-core* samples are at the fringes of clusters. A *core* sample is such that there are 'min_samples' other samples with a distance less than ϵ from it, for some $\epsilon > 0$ (Pedregosa et al., 2011). The HDBSCAN mechanism performs clustering for various ϵ values and the most stable clustering is chosen.

The default metric for HDBSCAN is Euclidean distance. To use cosine similarity, we turn it into a distance function (McInnes et al., 2017):

$$\|x - y\| = \sqrt{2 \times (1 - CosSim(x, y))}\ , \qquad (1)$$

where x, y are unit vectors, i.e., $\|x\| = \|y\| = 1$ (Manning, 1999). Since cosine similarity does not depend on vectors' magnitude, only on the angle between the two vectors, we first turned every embedding e_i to a unit vector $\frac{e_i}{\|e_i\|}$ and then applied the HDBSCAN on a pre-computed metric matrix consisting of all pairwise distances between all embeddings in the dataset using formula (1).

In contrast to the KMeans, HDBSCAN can find clusters with varied densities and clusters may have non-convex shapes.

4.3 Metrics for Comparing Clusters

To compare the similarity between the KPs and the cluster assignments of the KMeans/HDBSCAN, we used Adjusted Rand Index (ARI) (Vinh et al., 2009). In ARI, similarity is interpreted as the number of pairs of items on which the clusterings agree, adjusted for the amount of chance agreement. Let D be a dataset containing n items that are classified into m clusters by clustering C and, independently, into k clusters by clustering E. For a pair of items $(i_1, i_2) \in D$, C and E agree on it iff i_1 and i_2 are either (1) assigned to the same cluster in both C and E (let's say there are a such pairs), or (2) assigned to different clusters in both C and E (let's say there are b such pairs). Now, $a + b$ is the number of agreements between C and E. The ARI index is given by:

$$RI = \frac{a+b}{\binom{n}{2}} \quad ; \quad ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]} \quad ,$$

where $E[RI]$ is the expected RI for some random label assignment (Vinh et al., 2009), and $max(RI)$ equals to 1. The ARI values range from -1 to 1, where 1 indicates perfect agreement, and -1 indicates complete disagreement (Hubert and Arabie, 1985). Since each student response in our dataset is labeled with its KP, we evaluated the ARI for each clustering assignment, i.e., KMeans and HDBSCAN, compared to the KPs. This yields a *global* comparison between the KPs and each clustering assignment.

To evaluate the 'discoverability' of each KP, we also conducted ***by-KP analysis***, applying a retrieval paradigm and considering each cluster as an attempt to retrieve each of the KPs. We calculate *recall*, *precision*, and *F1* score using a contingency matrix $A = (a_{mn})_{1 \le m \le k,\ 1 \le n \le f}$ where rows are the KPs $k = 6, 7$, and columns are the unsupervised clusters C_n found by KMeans or HDBSCAN, $f = \#fitted\_clusters$;

$$a_{mn} = \sum_{1 \le m \le k,\ 1 \le n \le f} \left| \{x : \ x \in KP_m \cap C_n\} \right|$$

the cell a_{mn} in the matrix A counts the number of members of KP$_m$ that fell in cluster C$_n$. The precision of retrieval of KP$_m$ using cluster C$_n$ is $P_{mn} = \frac{a_{mn}}{|C(n)|}$; the recall is $R_{mn} = \frac{a_{mn}}{|KP(m)|}$. F1 score is $F_{mn} = \frac{2 \cdot P_{mn} \cdot R_{mn}}{P_{mn} + R_{mn}}$, indicating the extent to which we were able to retrieve KP$_m$ using the emergent cluster C$_n$.

5 Results

5.1 RQ1: Correspondence between embedding-based clusters and theory-based Knowledge Profiles

5.1.1 Global alignment between the clusterings

As described in Section 4, we evaluated the agreement between clusterings that were computed from the embeddings using two cluster analysis methods: KMeans and HDBSCAN. As we were interested in upper-bounding the discoverability of the KPs by both algorithms, we “helped" them with additional information (the number of clusters to the KMeans algorithm, and allowing the HDBSCAN to grid search for ‘good’ hyperparameters). With k equals the number of KPs per item (six for Q1 and seven for Q2), the resulting ARIs for the KMeans were **0.122** and **0.191**, for Q1 and Q2, respectively. For the HDBSCAN algorithm, we conducted a grid search for two parameters: $min\_cluster\_size$, i.e., the minimum number of samples in a cluster (values:$\{3, 4, 5, 10, 15, 20, 30, 40\}$), and $min\_samples$, i.e., the number of samples in a neighborhood for a point to be considered as a core point (values: $\{1, 2, 3, 4, 5\}$). We report the best-performing combination in terms of ARI: **0.037** for Q1 (with $min\_cluster = 5$, $min\_samples = 2$), and **0.038** for Q2 (with $min\_cluster = 3$, $min\_samples = 3$). Based on these results, we conclude that the clusters discovered by the KMeans had low agreement with the KPs, and the clusters discovered by the HDBSCAN had negligible agreement with the KPs.

5.1.2 Discoverability of specific KPs

We further investigated the clusters’ matching quality by calculating the F1 score *per KP* for each of Q1 and Q2. For KMeans, the results show good retrieval of KP1, the cluster with the highest-quality responses – $F1 = 0.60,\ 0.67$ for items Q1 and Q2 respectively – but much worse retrieval of the other KPs, with maximal $F1 = 0.40$ for KP6 in Q1 and $F1 = 0.47$ for KP2 in Q2. The clustering results in terms of contingency tables and F1 Scores are presented in Tables 1 to 4, with KPs as rows and columns as fitted clusters.The maximum F1 scores per profile are shaded in gray.

The evaluation of HDBSCAN clusters mirrored that of KMeans, showing better retrieval of KP1 – $0.43, 0.46$ F1 scores for Q1 and Q2 – than of any other profile, with maximal $F1 = 0.36$ for KP6 in Q1 and $F1 = 0.29$ for KP2 in Q2. We observe that, overall, results are worse for HDBSCAN than for KMeans. The clustering results in terms of contingency tables and F1 Scores are presented in Tables 5 to 8, with KPs as rows and columns as fitted clusters. The maximum F1 scores are shaded in gray.

We then considered the possibility that more coarse-grained profiles might emerge from the clustering than the detailed KPs. To this end, we tried different options for grouping KPs and calculated the F1 scores between fitted clusters and the grouped KPs. The best results show an emergent pattern consistent in both items Q1 and Q2, where one cluster consists of the higher-quality responses (KP1-KP4) an is retrievable with $F1 = 0.72, 0.74$, and the other cluster consists of lower-quality responses (KP5-KP7), retrievable with $F1 = 0.52, 0.45$.

We tried this approach with the KMeans as well, but the samples scattering across fitted clusters there did not exhibit meaningful patterns.

5.2 RQ2: How well are the KPs represented in the embeddings?

To answer this question, we first analyzed, descriptively, the level of similarity between the embeddings within each KP, and between KPs. We then conducted statistical tests to verify that the observed patterns are statistically robust.

Within KP similarity. To analyze the level of similarity within each KP, we computed the pairwise cosine-similarity between all pairs in that KP. Tables 9 and 10 show the results, with KPs as rows and fitted clusters as columns. Within-KP similarities are in the diagonals. Since the pairwise cosine similarity values are not normally distributed[1], we report medians. The results show that KP1’s embeddings (highest quality responses) have the highest density; as the quality of a response goes down, so does its similarity to other responses with the same pattern of error.

Between-KP similarity. As can be further seen in Tables 9 and 10, for both items, for every $i > 1$ the embeddings of KP$_i$ responses tend to be more similar to the embeddings of KP1 than to embeddings of their own KP (the bolded values in the first row are the largest in each column). This means that erroneous responses of various types are more similar to the correct responses than to those with the same pattern of error.

Hypothesis testing. Next, we conducted statistical tests to confirm that i) the distribution of the embeddings in each KP are indeed different and that ii) the cosine similarity within each KP is correlated with the responses quality.

i) A Kruskal-Wallis H-test confirmed that at least one of the medians for the different KPs is signif-

[1] The two-sided Kolmogorov-Smirnov test: test statistic = 0.5, $p < 0.001$

	A	B	C	D	E	F
KP1	0	102	0	0	5	24
KP2	3	39	0	0	12	37
KP3	3	22	0	0	13	65
KP4	13	28	0	0	26	39
KP5	12	14	0	0	36	50
KP6	31	5	3	16	55	16

Table 1: Q1 KMeans contingency matrix.

	A	B	C	D	E	F
KP1	.00	.60	.00	.00	.04	.13
KP2	.04	.26	.00	.00	.10	.23
KP3	.04	.14	.00	.00	.10	.39
KP4	.15	.18	.00	.00	.21	.23
KP5	.14	.09	.00	.00	.28	.29
KP6	.33	.03	.05	.23	.40	.09

Table 2: Q1 KMeans F1 Scores.

	A	B	C	D	E	F	G
KP1	10	0	2	13	13	127	0
KP2	51	0	22	13	4	30	0
KP3	11	0	10	21	14	28	0
KP4	12	0	18	25	16	13	5
KP5	5	0	16	23	29	6	1
KP6	3	0	29	8	8	6	10
KP7	4	9	20	11	8	3	12

Table 3: Q2 KMeans contingency matrix.

	A	B	C	D	E	F	G
KP1	.08	0	.01	.09	.10	.67	0
KP2	.47	0	.19	.11	.04	.18	0
KP3	.12	0	.10	.21	.16	.19	0
KP4	.13	0	.17	.25	.18	.09	.09
KP5	.06	0	.16	.24	.34	.04	.02
KP6	.04	0	.32	.09	.10	.04	.22
KP7	.05	.24	.22	.12	.10	.02	.25

Table 4: Q2 KMeans F1 Scores.

icantly different from the others, for both Q1 and Q2 (Q1: $statistic = 338.435, p < 0.001$; Q2: $statistic = 295.019, p < 0.001$). A follow-up Dunn's post-hoc analysis indicated that the within-KPs similarities differ significantly across all KP pairs, for both Q1 and Q2, with $p < 0.001$. This indicates that embeddings of responses from different KPs have different distributions. Moreover, the embeddings of high-quality responses are highly dense, while embeddings of low-quality responses are more scattered in the vector space.

ii) To show that the cosine similarities within KPs are significantly correlated with the responses' quality, we calculated for every sample $x \in KP(i)$ its cosine similarity to $KP(i)$ centroid $c(i)$, where $c(i)$ is the average embedding component-wise of the embeddings in $KP(i)$, i.e.,

$$CosSim(x, c(i)) \quad \forall x \in KP(i).$$

We then calculated the Spearman correlation between all the similarities values $\cup_{i=1}^{k} \{CosSim(x, c(i)) \ : \ x \ \in \ KP(i)\}$ where $k = 6, 7$ for Q1, Q2 respectively, and the ordinal variable of the KPs' index, where lower index represents higher-quality responses. The Spearman correlation coefficient and its p-values are:

$$r_{Q1} = -0.686, \ p < 0.001$$

$$r_{Q2} = -0.633, p < 0.001$$

indicating a strong correlation (Xiao et al., 2016) between the quality of a 'family of responses' (KP) and the within-family similarity.

6 Discussion and Conclusion

Our data consists of 669 high-school student responses to two typical constructed response items in high-school biology. The responses were human graded according to an analytic rubric that is based on the Causal-Mechanical explanation framework (Ariely et al., 2023), transforming each response to a binary vector that encodes the grading according to the rubric categories. Previous work demonstrated that applying cluster analysis (KMeans) to these vectors, which result from a process that applies a theoretical assessment framework to concrete context by human experts, yields stable clusters that reveal pedagogically meaningful knowledge profiles, which were validated in several ways (Ariely et al., 2024). (For more details, see Section 3.) We reasoned that given the successful performance of pre-trained LLMs on a variety of education-related meaning-intensive tasks (Schleifer et al., 2023; Wambsganss et al., 2023; Riordan et al., 2020; Sung et al., 2019), and previous work that applied this specifically to profile discovery (Martin et al., 2023; Wulff et al., 2022), we want to evaluate whether unsupervised profile discovery that is not aided by human knowledge works sufficiently well to be applied out-of-

	A	B	C	D
KP1	19	0	112	0
KP2	29	0	62	0
KP3	36	0	67	0
KP4	49	0	57	0
KP5	61	0	51	0
KP6	71	7	43	5

Table 5: Q1 HDBSCAN contingency matrix.

	A	B	C	D
KP1	.10	0	.43	0
KP2	.16	0	.26	0
KP3	.20	0	.27	0
KP4	.26	0	.23	0
KP5	.32	0	.20	0
KP6	.36	.11	.17	.08

Table 6: Q1 HDBSCAN F1 Scores.

	A	B	C	D
KP1	23	0	142	0
KP2	39	0	81	0
KP3	27	0	57	0
KP4	30	0	56	3
KP5	32	0	48	0
KP6	33	0	31	0
KP7	32	3	32	0

Table 7: Q2 HDBSCAN contingency matrix.

	A	B	C	D
KP1	.12	0	.46	0
KP2	.23	0	.29	0
KP3	.18	0	.21	0
KP4	.20	0	.21	.07
KP5	.22	0	.18	0
KP6	.24	0	.12	0
KP7	.23	.09	.12	0

Table 8: Q2 HDBSCAN F1 Scores.

the-box.

The results of RQ1 reveal that two distinct common unsupervised clustering techniques largely failed to discover the ‘gold’ KPs from the pre-trained LLM embeddings. Inasmuch as a weak relationship with the knowledge profiles was exhibited by KMeans clusters (ARI of 0.12-0.19), our retrieval-based analysis per profile showed that KP1, the profile that captures the correct responses, was the most discoverable profile, with F1-scores of 0.60/0.67 (on Q1/Q2) for retrieving members of KP1 using the best-aligned emergent cluster. Thus, had the emergent clusters been used as a basis for feedback, only the correct responses would have received pedagogically cogent feedback, since responses belonging to low-knowledge KPs are all intermixed in the emergent clusters. This phenomenon was consistent across two items – Q1 and Q2 – that were analyzed separately.

In an attempt to account for both the failure of overall profile discovery based on pre-trained LLM embeddings and for the bias towards the correct responses exhibited by the emergent clusters, we turned ‘upstream’ to inspect how the KPs are represented by the embeddings.

We found that the lower the knowledge level of the profile, the less similar to each other its members are in the embeddings space. It is this property that we refer to as the **Anna Karenina principle**: Analogously to Tolstoy’s observation that happy families are similar to each other whereas each unhappy family is unhappy in its own way, we see that the correct responses are similar to each other, whereas incorrect responses differ more from each other the more incorrect (’unhappy’) they are (strong correlation of $r_1 = -0.686, r_2 = -0.633$ for Q1/Q2). One could say that Tolstoy considered all unhappy families as an undifferentiated mass; presumably, if classified by their specific source of unhappiness (by family therapists, say), profiles would have probably emerged. In our case, the incorrect responses are grouped by teachers according to the type of problem they exhibit; however, within-profile similarities are still lower than those of correct responses and drift further apart the bigger the problems. The lower density of the poor-knowledge profiles may be one reason that inhibits their downstream discovery.

Further analysis suggests that the privileged status of ‘happy families’ (correct responses) extends beyond their higher density. We also found that while an average correct response is most similar to another correct response, an average incorrect response is closer in the embeddings space to a correct response than to a member of its own profile (Tables 9 and 10). That is, in some sense, the correct responses are the center of the universe, whereas the incorrect responses drift around them in non-convex formations. The non-convexity of the lower-knowledge profiles may be another inhibitor of their downstream discoverability. Taken together, the ‘classic’ Anna Karenina property and

KP	1	2	3	4	5	6
1	**.920**	**.910**	**.900**	**.899**	**.883**	**.852**
2		.903	.896	.889	.880	.849
3			.897	.890	.883	.852
4				.877	.871	.837
5					.874	.845
6						.755

Table 9: Q1 pairwise cosine similarity median per KP.

KP	1	2	3	4	5	6	7
1	**.916**	**.891**	**.896**	**.886**	**.873**	**.857**	**.837**
2		.876	.870	.860	.854	.840	.820
3			.881	.870	.862	.844	.828
4				.866	.860	.843	.822
5					.861	.843	.823
6						.824	.796
7							.764

Table 10: Q2 pairwise cosine similarity median per KP.

the strong version that puts the correct responses in the center suggest an explanation for both the overall discovery failure observed downstream and for the bias in favor of correct responses exhibited by the emergent clusters.

Based on our results, pre-trained representations may not lend themselves to making the necessary distinctions to support pedagogical decisions such as providing formative feedback that targets specific errors in student reasoning. In particular, our results show a case where the representations are not sufficiently nuanced to allow commonly used clustering methods to identify any error-based profiles, only the profile of the correct responses. Since it is the students who gave the incorrect responses who are in most need of targeted formative feedback, the bias in favor of correct responses is especially counter-productive. Thus, our results tell a cautionary tale about using emergent properties of student response data built over pre-trained embeddings without domain- and task-specific tuning, and without human supervision.

6.1 Limitations

It is possible that other clustering approaches could have revealed clusters that are more similar to the 'gold' ones. However, given that despite the large difference between KMeans and HDBSCAN's algorithmic approach, they were quite consistent in both demonstrating poor overall agreement and being biased towards discovering the best KP, we believe that reaching results that are qualitatively different from another clustering method is unlikely. It is also possible that emergent clusters do correspond to an alternative meaningful partition of the responses into groups, but that partition is not what educators see when they analyze student responses.

The AlephBERT model used in this paper is state-of-the-art for Hebrew, but it has a smaller number of parameters compared to the most recent LLMs for English. It is possible that with more advanced LLM technology, the LLM representations of student responses will be more nuanced; we will revisit our analyses with larger Hebrew LLMs when available.

Due to the monolingual nature of our current data, we have experimented with one language only. Work is underway to collect comparable student response data in Arabic.

Ethics statement

We acknowledge that the work is conformant with the ACL Code of Ethics. The research and its data collection procedures were approved by the Institutional Review Board and the Ministry of Education. The instrument was administered to the students as part of the regular instruction of the topic, based on the teachers' decision to use it as part of the teaching routine (the instrument was published in teachers' forums), with teacher and school principal approval that response data will be used for research.

The goal of this research is to better understand the relations between pre-trained LLM-based representations of student responses to open-ended questions in science, and representations of these responses according to theory-driven rubrics applied by human experts. We study to what extent conceptually/pedagogically similar responses tend to maintain their proximity in the embedding space as well, and the impact of deviations from this property on downstream analysis. What makes this especially relevant to Ethics is our finding that the weaker students are the ones whose responses suffer the most from representation mismatches between the two representation spaces. This limits the ability to automatically cater to these students – the ones who are in the highest need for personalized guidance – with formative feedback that matches the gaps in their reasoning. By identifying and naming this phenomenon ('the Anna Karenina principle' in automated short answer evaluation), we hope to start a discussion on the means to both estimate its prevalence and to address it.

We demonstrate the Anna Karenina principle on two tasks with one pre-trained model. It is possible that results will look different with other tasks and other large language models. There is a potential danger of over-generalization based on our results, whereby large language models, as a species, so to speak, would be thought to suffer from the Anna Karenina principle and their off-the-shelf use would be avoided in learner-focused applications. This, in turn, could hamper development of useful LLM-based applications to support learners. We believe that the best course of action is to continue the study of the principle in order to improve our understanding of what kind of models are likely to exhibit the problem and for what kind of task, as well as how to diagnose and correct it, ideally without recourse to a large human-tagged dataset. In parallel, future ethics-focused research could investigate whether weaker learners should be a protected category in educational applications, akin to demographic categories like race or gender, by investigating evidence of harm differentially wrought on such learners through technology that does not cater sufficiently precisely to their needs.

Data from this research cannot be shared publicly due to privacy regulations, but may be provided for research purposes, along with its analysis code, subject to the necessary approvals.

Acknowledgments

This study was partially supported by the Israeli Science Foundation (ISF) grant number 851/22.

References

Nico Andersen, Fabian Zehner, and Frank Goldhammer. 2023. Semi-automatic coding of open-ended text responses in large-scale assessments. *Journal of Computer Assisted Learning*, 39(3):841–854.

Moriah Ariely, Tanya Nazaretsky, and Giora Alexandron. 2022. Personalized automated formative feedback can support students in generating causal explanations in biology. In *Proceedings of the 16th International Conference of the Learning Sciences-ICLS 2022, pp. 953-956*. International Society of the Learning Sciences.

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.

Moriah Ariely, Tanya Nazaretsky, and Giora Alexandron. 2024. Causal-mechanical explanations in biology: Applying automated assessment for personalized learning in the science classroom. *Journal of Research in Science Teaching*.

Perpetual Baffour, Tor Saxberg, and Scott Crossley. 2023. Analyzing bias in large language model solutions for assisted writing feedback tools: Lessons from the feedback prize competition series. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 242–246.

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.

Angela Bovo, Stéphane Sanchez, Olivier Héguy, and Yves Duthen. 2013. Clustering moodle data as a tool for profiling students. In *2013 Second international conference on E-Learning and E-Technologies in education (ICEEE)*, pages 121–126. IEEE.

Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Yang Trista Cao, Yada Pruksachatkun, Kai-Wei Chang, Rahul Gupta, Varun Kumar, Jwala Dhamala, and Aram Galstyan. 2022. On the intrinsic and extrinsic fairness evaluation metrics for contextualized language representations. In *Proceedings of the ACL*, pages 561–570.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*.

Nancy J Fellows. 1994. A window into thinking: Using student writing to understand conceptual change in science learning. *Journal of Research in Science Teaching*, 31(9):985–1001.

Libby F Gerard and Marcia C Linn. 2016. Using automated scores of student essays to support teacher guidance in classroom inquiry. *Journal of Science Teacher Education*, 27(1):111–129.

Seraphina Goldfarb-Tarrant, Rebecca Marchant, Ricardo Muñoz Sánchez, Mugdha Pandya, and Adam Lopez. 2021. Intrinsic bias metrics do not correlate with application bias. In *Proceedings of the ACL*, pages 1926–1940.

Seraphina Goldfarb-Tarrant, Eddie Ungless, Esma Balkir, and Su Lin Blodgett. 2023. This prompt is measuring <mask>: evaluating bias evaluation in language models. In *Findings of the ACL*, pages 2209–2225.

John Hattie and Helen Timperley. 2007. The power of feedback. *Review of Educational Research*, 77(1):81–112.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of Classification*, pages 193–218.

Masahiro Kaneko, Danushka Bollegala, and Naoaki Okazaki. 2022. Debiasing isn't enough! – on the effectiveness of debiasing MLMs and their social biases in downstream tasks. In *Proceedings of COLING*, pages 1299–1310.

Yaman Kumar, Swati Aggarwal, Debanjan Mahata, Rajiv Ratn Shah, Ponnurangam Kumaraguru, and Roger Zimmermann. 2019. Get IT scored using AutoSAS — an automated system for scoring short answers. In *Proceedings of AAAI*, pages 9662–9669.

Tai Le Quy, Gunnar Friege, and Eirini Ntoutsi. 2023. A review of clustering models in educational data science toward fairness-aware learning. In Alejandro Peña-Ayala, editor, *Educational Data Science: Essentials, Approaches, and Tendencies: Proactive Education based on Empirical Big Data Evidence*, pages 43–94. Springer Nature Singapore, Singapore.

Tingting Li, Emily Reigh, Peng He, and Emily Adah Miller. 2023. Can we and should we use artificial intelligence for formative assessment in science? *Journal of Research in Science Teaching*, 60(6):1385–1389.

Zhaohui Li, Yajur Tomar, and Rebecca J. Passonneau. 2021. A semantic feature-wise transformation relation network for automatic short answer grading. In *Proceedings of EMNLP*, pages 6030–6040.

Stuart Lloyd. 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137.

Schutze H. Manning, C. D. 1999. *Foundations of statistical natural language processing*. MIT Press.

Paul P Martin, David Kranz, Peter Wulff, and Nicole Graulich. 2023. Exploring new depths: Applying machine learning for the analysis of student argumentation in chemistry. *Journal of Research in Science Teaching*.

Mihai Masala, Stefan Ruseti, Mihai Dascalu, and Ciprian Dobre. 2021. Extracting and clustering main ideas from student feedback using language models. In *Proceedings of AIED*, pages 282–292. Springer.

Leland McInnes, John Healy, and Steve Astels. 2017. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11).

Tomoya Mizumoto, Hiroki Ouchi, Yoriko Isobe, Paul Reisert, Ryo Nagata, Satoshi Sekine, and Kentaro Inui. 2019. Analytic score prediction and justification identification in automated short answer scoring. In *Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 316–325.

Ross H Nehm and Hendrik Haertig. 2012. Human vs. computer diagnosis of students' natural selection knowledge: testing the efficacy of text analytic software. *Journal of Science Education and Technology*, 21:56–73.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

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.

Juan L Rastrollo-Guerrero, Juan A Gómez-Pulido, and Arturo Durán-Domínguez. 2020. Analyzing and predicting students' performance by means of machine learning: A review. *Applied Sciences*, 10(3):1042.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks. In *Proceedings of EMNLP*, pages 3982–3992.

Brian Riordan, Sarah Bichler, Allison Bradford, Jennifer King Chen, Korah Wiley, Libby Gerard, and Marcia C. Linn. 2020. An empirical investigation of neural methods for content scoring of science explanations. In *Proceedings of the 15th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 135–144.

Wesley C Salmon. 2006. *Four decades of scientific explanation*. University of Pittsburgh press.

Abigail Gurin Schleifer, Beata Beigman Klebanov, Moriah Ariely, and Giora Alexandron. 2023. Transformer-based Hebrew NLP models for short answer scoring in biology. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 550–555.

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 ACL*, pages 46–56.

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 ACL*, pages 1703–1714.

Chul Sung, Tejas Indulal Dhamecha, and Nirmal Mukhi. 2019. Improving short answer grading using transformer-based pre-training. In *Proceedings of AIED*, pages 469–481.

Charissa Tansomboon, Libby F Gerard, Jonathan M Vitale, and Marcia C Linn. 2017. Designing automated guidance to promote productive revision of science explanations. *International Journal of Artificial Intelligence in Education*, 27:729–757.

Stephen E Toulmin. 2003. *The uses of argument*. Cambridge University Press.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2009. Information theoretic measures for clusterings comparison: Is a correction for chance necessary? In *Proceedings of ICML*, pages 1073–1080.

Thiemo Wambsganss, Xiaotian Su, Vinitra Swamy, Seyed Neshaei, Roman Rietsche, and Tanja Käser. 2023. Unraveling Downstream Gender Bias from Large Language Models: A Study on AI Educational Writing Assistance. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10275–10288.

Peter Wulff, David Buschhüter, Andrea Westphal, Lukas Mientus, Anna Nowak, and Andreas Borowski. 2022. Bridging the gap between qualitative and quantitative assessment in science education research with machine learning—a case for pretrained language models-based clustering. *Journal of Science Education and Technology*, 31(4):490–513.

Chengwei Xiao, Jiaqi Ye, Rui Máximo Esteves, and Chunming Rong. 2016. Using spearman's correlation coefficients for exploratory data analysis on big dataset. *Concurrency and Computation: Practice and Experience*, 28(14):3866–3878.

Appendix 1

Item	Text
Smoking item	The smoke from cigarettes contains several harmful substances, including the gas carbon-monoxide (CO). CO is released from cigarettes while smoking, and has a stronger tendency than oxygen to bind to Hemoglobin. Explain how high levels of CO make it difficult for smokers to exercise.
Anemia item	A person was found to have low levels of red blood cells in his blood test (anemia). This person complained to his doctor about weakness and difficulty to exercise. Explain how low levels of red blood cells make it difficult for people with anemia to exercise.

Table 11: The constructed response items (reproduced from Ariely et al. (2024); original responses are in Hebrew).

Cluster description	Exemplifying response
Full explanations: All/most of the conceptual components and the underlying causal relations are present.	"Red blood cells bind oxygen and transfer it in the bloodstream, from the lungs where it is absorbed, to all the cells of the body. A low amount of red blood cells in the body leads to the transfer of less oxygen to the body's cells. Since oxygen is one of the reactants in the process of cellular respiration - the energy production process, less oxygen reaching the cells leads to damage to this process. Thus, less available energy is produced in the body's cells and this impairs their function, which leads to fatigue and difficulty in performing physical activity." (Anemia item)
Gaps in causal connections: All/most of the conceptual components are present, but all/most of the causal relations are missing.	"The CO binds to the red blood cell instead of the oxygen and thus oxygen does not reach the cells of the body and then cellular respiration does not occur and the body cannot produce energy and thus it stops physical activity due to lack of energy." (Smoking item)
Specific sequential stages are missing and causal relations are often missing.	"CO gas is known to bind to Hemoglobin with a stronger tendency than oxygen. When CO binds to Hemoglobin, it takes the oxygen's place, so much less oxygen is transported from place to place and enters the cells. Lack of oxygen in the cells leads to less production of ATP molecules. Since energy is required for physical exercise, the result is that the person gets tired quickly and has difficulty exercising." (Smoking item)
Many sequential stages are missing and causal relations are often missing as well.	"Red blood cells carry the oxygen (because of Hemoglobin). When there is anemia, then there is a low amount of red blood cells and thus a low amount of oxygen reaches the muscles." (Anemia item)
No explanation: All/most of the sequential stages and the underlying causal relations are missing/irrelevant responses.	"I don't know" "Anemic people are tired because they have few red blood cells." (Anemia item)

Table 12: Examples of student responses and their classification into the KPs that were derived from the expert scoring according to the theory-driven rubric (reproduced from Ariely et al. (2024); original responses are in Hebrew).

Assessing Student Explanations with Large Language Models Using Fine-Tuning and Few-Shot Learning

Dan Carpenter[1], **Wookhee Min**[1], **Seung Lee**[1],
Gamze Ozogul[2], **Xiaoying Zheng**[2], **James Lester**[1]

{dcarpen2, wmin, sylee, lester}@ncsu.edu
gozogul@indiana.edu, zheng12@iu.edu
[1] North Carolina State University
[2] Indiana University

Abstract

The practice of soliciting self-explanations from students is widely recognized for its pedagogical benefits. However, the labor-intensive effort required to manually assess students' explanations makes it impractical for classroom settings. As a result, many current solutions to gauge students' understanding during class are often limited to multiple choice or fill-in-the-blank questions, which are less effective at exposing misconceptions or helping students to understand and integrate new concepts. Recent advances in large language models (LLMs) present an opportunity to assess student explanations in real-time, making explanation-based classroom response systems feasible for implementation. In this work, we investigate LLM-based approaches for assessing the correctness of students' explanations in response to undergraduate computer science questions. We investigate alternative prompting approaches for multiple LLMs (i.e., Llama 2, GPT-3.5, and GPT-4) and compare their performance to FLAN-T5 models trained in a fine-tuning manner. The results suggest that the highest accuracy and weighted F1 score were achieved by fine-tuning FLAN-T5, while an in-context learning approach with GPT-4 attains the highest macro F1 score.

1 Introduction

Interactivity is critical to learning (Blasco-Arcas et al. 2013; Herppich et al. 2016). It has been widely demonstrated that by increasing interactivity in the classroom, we can significantly improve students' learning outcomes (Beauchamp and Kennewell 2010; Mayer et al. 2009). Student-teacher interaction is one of the most influential factors in learning (Beauchamp and Kennewell 2010), and when classrooms are interactive, students become more engaged, more participative, and are more motivated to learn (Bachman and Bachman 2011; Barnett 2006; Caldwell 2007). In addition, interactivity can improve comprehension and lead to improved learning (Freeman et al. 2014). Despite these benefits, many STEM classrooms use lectures as the primary method of instruction. The lack of interactivity poses serious issues in undergraduate education (Freeman et al. 2014), and large class sizes can inhibit meaningful exchanges between instructors and students in traditional classrooms (Caldwell 2007). The passive nature of lectures is particularly problematic in STEM courses, as research shows that undergraduate students in classes that use a traditional lecture format are much more likely to fail than students in classes that use a more active learning method (Freeman et al. 2014).

Classroom response systems have been touted as a potential solution to this problem. These systems capture and grade student responses to multiple choice questions posed by instructors during lectures. Each student submits a response using a handheld transmitter (a "clicker"), and software on the instructor's computer records, grades, and displays students' answers for the class to view. While research has shown that classroom response systems can promote student engagement and facilitate the learning of factual knowledge (Campbell and Mayer 2009; Hunsu et al. 2016), studies have also shown that "clickers" are less effective for promoting deep and meaningful learning. In fact, traditional classroom response systems may actually obstruct students from developing a conceptual understanding of concepts and principles, particularly for novice students (Shapiro et al. 2017). Because students simply select an answer from a list of choices, "clickers" do not enable students to construct or generate their own responses to questions, which is a key component of active and constructive learning (Chi and Wylie 2014).

Decades of research have shown that self-explanation has a significant impact on student learning (Chi et al., 1994; Fonseca and Chi 2011). By explaining concepts and examples to themselves as they learn, students trigger the self-explanation effect, where they actively probe their own understanding and address gaps in their knowledge. Enabling students to generate short-answer textual explanations to prompts posed by instructors during lectures could open a rich communication channel between instructors and students. Eliciting self-explanations from students has the potential to yield substantial learning benefits for students in undergraduate STEM classrooms, and it has been widely demonstrated that self-explanation helps students learn much more effectively than students who do not engage in self-explanation (Chi et al., 1994; Fonseca and Chi 2011; Johnson and Mayer 2010; Roy and Chi 2005). Because self-explanation requires students to explain concepts to themselves in their own words, they learn much more deeply. However, despite the great potential offered by self-explanation for promoting learning, students in undergraduate STEM classrooms often have limited opportunities to engage in this type of active and constructive learning activity due to limited class time for discussion. Similarly, instructors have limited time to assess students' self-explanation responses and provide formative and timely feedback during lectures.

In this paper, we present a large language model-based approach that automatically assesses students' written responses. We investigate the performance of four Transformer-based large language models—Llama 2 (Touvron et al. 2022), GPT-3.5 (OpenAI 2023), GPT-4 (OpenAI 2023), and FLAN-T5 (Chung et al. 2022)—in assessing the correctness (i.e., fully correct, partially correct, and incorrect) of student self-explanations to undergraduate computer science questions. These explanations were collected from undergraduate students, including those who participated in an undergraduate course using the EXPLAINIT system we have developed. Our findings suggest that FLAN-T5 demonstrates high performance in terms of accuracy and weighted F1, when fine-tuned using a prompt that includes information taken from a grading rubric in combination with an exemplar response provided by the instructor. However, we also find that the highest macro F1 score is achieved by GPT-4 in a few-shot learning setting, where examples of only ten students' explanation responses are provided without any additional information from a rubric or an exemplar response. We discuss the tradeoffs between these models and the implications of our research for practical applications of LLM-based explanation assessment in classroom response systems.

2 Related Work

It has been found that students explaining concepts to themselves has a profound effect on learning. Known as the self-explanation effect (Chi et al. 1994; Fonseca and Chi 2011; Sidney et al. 2015), the result of self-explanation goes beyond simply rehearsing information: it requires students to express concepts in their own words, relate concepts to prior knowledge, make inferences, integrate information with prior knowledge, and monitor and repair faulty knowledge. Thus, self-explanation is a deeply constructive activity (Roy and Chi 2005). The significant learning gains associated with self-explanation have been demonstrated in a wide range of STEM disciplines including computer science (Pirolli and Recker, 1994), engineering (Johnson and Mayer 2010), chemistry (Crippen and Earl 2007), algebra (Atkinson et al. 2003), biology (McNamara 2004), physics (Chi et al. 1994), and physiology (Butcher 2006). Our EXPLAINIT classroom response system leverages the self-explanation effect to improve STEM classroom learning.

Widely known as "clickers," classroom response systems have emerged as a tool to bridge the gap between students and instructors and to make lectures more interactive. Used by millions of students, classroom response systems allow students to anonymously respond to multiple choice questions presented during lectures. Research has shown that students appreciate the ability to compare their own answers to those of their peers, receive immediate feedback, and test their knowledge, and that "clickers" can increase student interactivity during lectures (Freeman et al.

2014; Hunsu et al. 2016; Kay and LeSage 2009). However, studies have also shown that clickers fail to promote deep and meaningful learning, which can be particularly problematic for students in STEM classes who are required to conceptually understand important concepts, relationships, and theories to effectively solve problems (Shapiro et al. 2017). Closely related to our work, commercial classroom response systems have been explored in various classroom settings. These systems typically support students through classroom discussions, questions, and assignments, and they support instructors with features for course material creation and assessment, which are incorporated with learning management systems. While they provide a range of functionalities required for a classroom response system, such as the ability to pose various types of questions (e.g., multiple choice, fill-in-the-blank, short answer questions), their automated assessment is typically limited to multiple choice and fill-in-the-blank types of questions that accept a predetermined set of answers, while they require a manual assessment process for other types of questions.

Deep learning-based language models such as BERT (e.g., Liu et al. 2019), FLAN-T5 (e.g., Chung et al. 2022), GPT (e.g., Brown et al. 2020), and Llama (Touvron et al. 2023) have been pivotal in the recent advancements in natural language processing (NLP; Torfi et al. 2020). In learning analytics, additional sources of training data, including data collected for free-response prompts (Rivera-Bergollo et al. 2022), text providing additional context for free-response prompts (Condor et al. 2021), response assessment rubrics (Condor et al. 2022), and synthetic data generated via data augmentation strategies (Lun et al. 2020), have effectively enhanced the training of NLP models, leading to improved predictive performance. NLP techniques have been used to accurately analyze student textual responses in the context of short-answer science assessment (Smith et al. 2019), student written reflections (Carpenter, Geden, et al. 2020), student-tutor dialogue (Carpenter, Emerson, et al. 2020), and student self-explanations (Chen and Wang 2022).

While previous work demonstrated considerable success with LLMs for short answer grading (Takano and Ichikawa 2022; Zhang et al. 2022) and short answer question generation (Moore et al. 2022), a research area that has seen limited exploration is assessing students' free-text explanations (Nicula et al. 2023). Building on recent advances in NLP and deep learning-based language modeling techniques, our work makes a novel contribution by investigating an approach to assess students’ self-explanations, collected from an undergraduate Artificial Intelligence course, utilizing large language models with fine-tuning and few-shot learning.

3 ExplainIt Classroom Response System

The ExplainIt classroom response system leverages the self-explanation effect and active, constructive, and interactive learning, along with state-of-the-art natural language processing, to significantly improve STEM undergraduate education. With a specific focus on computer science, biology, and physics, it aims to create highly engaging classroom learning experiences. ExplainIt offers the opportunity to fundamentally improve classroom dynamics by supporting both students and instructors. The system is designed to support both students and instructors in undergraduate STEM courses by analyzing and providing feedback on students’ explanations through an integrated five-step explanation feedback loop (Figure 1): (1) the instructor issues an explanation prompt, which appears in the ExplainIt app on students’ computing devices (e.g., laptops, tablets, phones); (2) students write free-text explanations ranging from a sentence to a short paragraph in the ExplainIt app on their computing devices; (3) ExplainIt automatically analyzes students’ explanations and provides real-time formative feedback to students individually in their apps; (4) ExplainIt provides a summary of correctness of student explanations to the instructor; and (5) the instructor makes “instructional pivots” by immediately tailoring pedagogy to respond to students’ explanations to improve student learning and engagement by focusing the lecture and classroom discussion on the most important elements of the course material. Collectively, these interactive explanation-based activities are designed to synergistically lead to improved student learning and promote greater student engagement in undergraduate STEM classrooms.

Our initial prototype of the ExplainIt classroom response system was implemented using a web-based application architecture to support enhanced scalability, where instructors

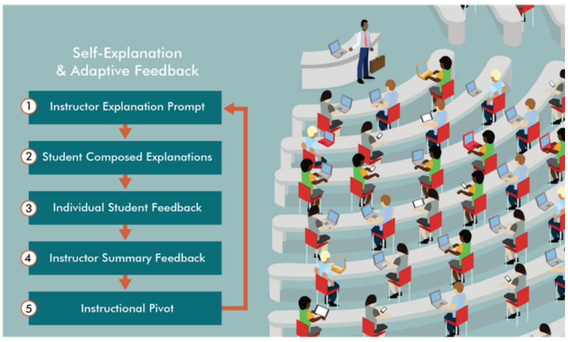

Self-Explanation
& Adaptive Feedback
1 Instructor Explanation Prompt
2 Student Composed Explanations
3 Individual Student Feedback
4 Instructor Summary Feedback
5 Instructional Pivot

Figure 1: The ExplainIt explanation-based classroom response system.

and students can choose their platforms of choice such as laptops and handheld devices, while the software enables real-time interactions with the user interfaces. The ExplainIt user interfaces consist of an Instructor Authoring Tool, Instructor Dashboard, and Student Explanation App. The streams of communication data are uploaded into a cloud-based database by the server. For data synchronization and analysis purposes, all interaction data is timestamped. To support these functionalities, we implemented the software modules to include APIs using the HTTP protocol. We use Microsoft's Azure cloud computing service to host our cloud-based services.

The Instructor Authoring Tool enables instructors to create or edit questions and an exemplar correct response per question. All questions and responses are categorized by their subjects and topics in the tool. All authored content is stored and accessed from the cloud, allowing the original instructor to reference their own created questions for future courses. The Instructor Dashboard presents the pool of questions per subject and topic, and it allows instructors to select and send questions to the Student Explanation App, so that students can view and interact with the questions in real-time during lectures. The Instructor Dashboard is also designed to display student-written responses and NLP assessment results in visual analytics. The Student Explanation App enables students to receive questions posed by instructors and write self-explanation responses to instructor-posed questions. When students submit their responses, the Student Explanation App taps into the Explanation Analyzer, which performs NLP-driven assessment of student explanations, generates tailored feedback to students, and dispatches analytical summaries to instructors through Instructor Dashboard. The Explanation Analyzer is in the development phase, and our findings about the Explanation Analyzer's NLP performance are presented in this paper.

4 Study and Data

This work uses data collected during a classroom pilot study of the ExplainIt system. The participants in the classroom study consisted of 36 consented undergraduate students enrolled in a Computer Science course focused on Artificial Intelligence. Thirty-two students completed a demographic pre-survey, and among them 8 indicated that they identified as female, 23 as male, and 1 preferred not to indicate gender identification. Participants ranged in age from 18 to 28 (M = 21.1, SD = 1.64). Of these participants, 40.6% were Asian, 50.0% were White, and 9.4% preferred not to answer.

Prior to using ExplainIt in the class, the instructor used the Instructor Authoring Tool to prepare a set of questions, each accompanied by an exemplar correct answer. These answers were presented to students immediately after they submitted their responses to the respective questions. The classroom implementation unfolded over 6 weeks within a single semester. Throughout this period, a total of 13 questions were sent to the class, eliciting 356 responses from 36 participants, which were utilized in our evaluation (Table 1).

Students' responses to the questions were labeled by two of the researchers, who are experts in computer science. First, a rubric item was constructed for each question that described the qualities of a correct, partially correct, or incorrect answer to the question. For example, the rubric for the question "In a neural network, what function is responsible for introducing non-linearity to the model?" indicated that a correct response should mention the term "activation function", a partially correct response might present an example of an activation function (e.g., "sigmoid") without explicitly mentioning the term "activation function", and that an incorrect response would not include any of this information. We also referenced instructor-provided exemplar answers to further refine the rubrics for each question. These were comprehensive and well-reasoned responses, serving as a representative correct answer to each question.

Then, based on the developed rubric, both researchers labeled twenty percent of the student

Question	Topic	Number of Questions Sent	Number of Student Responses
What does the term "deep" in deep learning refer to?	Deep Learning	1	28
What is the basic building block of a neural network called?	Deep Learning	1	27
In a neural network, what function is responsible for introducing non-linearity to the model?	Deep Learning	1	24
What is clustering in the context of machine learning?	Clustering	2	37
Name a commonly used algorithm for clustering and briefly describe how it works.	Clustering	2	34
What is the main difference between K-means and hierarchical clustering?	Clustering	2	37
The K-means algorithm may end up with different clustering results when the initial clustering centers are chosen differently. Yes or No?	Clustering	2	36
What is the "purity" of an external measure for cluster quality?	Clustering	1	22
What are support vectors in the context of SVMs	SVM	1	23
How does a soft-margin SVM differ from a hard-margin SVM?	SVM	1	23
Is it always better to use a soft-margin SVM to ensure model flexibility? Why?	SVM	1	22
Is an SVM more suitable for small datasets than large datasets? Why?	SVM	1	23
Can SVMs be used for both classification and regression tasks? Example?	SVM	1	20

Table 1: Descriptive statistics of questions sent during the classroom study.

responses. After one cycle of rubric refinement, a Cohen's Kappa of 0.702 was achieved, indicating substantial agreement (McHugh 2012). All labels that the annotators did not agree on were discussed and agreement on a single label was reached. Across all questions, 73% of explanations were labeled as correct, 22% were labeled as partially correct, and 5% were labeled as incorrect.

5 Method

We evaluated the performance of Llama 2 (Touvron et al. 2022), GPT-3.5 (OpenAI 2023), GPT-4 (OpenAI 2023), and FLAN-T5 (Chung et al. 2022) on the self-explanation assessment task. Large language models (LLMs) have been demonstrated to achieve state-of-the-art performance on many natural language processing tasks, with GPT-4 particularly excelling with few-shot prompting where training examples are integrated into the task description (OpenAI 2023). This enables GPT-4 to readily adapt to new tasks without re-training, avoiding the prohibitive cost of updating its extensive parameters. However, GPT models' proprietary nature and associated costs pose barriers to its educational adoption, such as EXPLAINIT.

To address this challenge, we also evaluated the performance of open-source models, FLAN-T5 and Llama 2. FLAN-T5 is an instruction-fine-tuned language model that has demonstrated competitive performance with other state-of-the-art models across a range of tasks when it was released (Chung et al. 2022). Llama 2 is an open-source pre-trained large language model that has demonstrated leading performance compared to other open-source models and performs similarly to GPT-3.5 on several tasks (Touvron et al. 2023). In this work, we investigate the performance of the base FLAN-T5 model (250M parameters) and Llama 2-7B, the smallest version of the model. These versions of FLAN-T5 and Llama 2 were selected due to their computational efficiency. For all models, default hyperparameters were used.

We investigated several different zero-shot and few-shot prompting approaches to evaluate the performance of Llama 2, GPT-3.5, and GPT-4 for automated assessment of students' self-explanation. As a baseline, these models were provided with instructions that described the task (i.e., "Please evaluate a student's explanation response to the following question.") in addition to the question and student response. Then, we systematically evaluated the impact on model performance of including the following information in the prompt: (a) rubric items for the current question, (b) an exemplar correct response provided by the instructor, and (c) other students' labeled responses to the current question. Prompts were constructed with all possible combinations of the different information elements, and model performance was evaluated for each combination.

For the prompts incorporating student self-explanation responses, 10-fold student-level cross-validation was used to prevent bias from students' individual writing styles and to ensure generalizability, avoiding data leakage in model evaluation. Additionally, this approach accurately represents the real-world scenario that will be faced when deploying EXPLAINIT in future classroom implementations, as the students interacting with the system will be new but the models will have access to past student's responses to each question. Due to LLM token limits and the per-token cost of proprietary models like GPT-4, we sampled ten responses from the training set for each cross-validation fold to include in the prompts rather than including the entire training set.

In comparison to Llama 2, GPT-3.5, and GPT-4, the FLAN-T5 base model's smaller parameter count facilitates easier and more cost-effective training. Given its sufficient size for fine-tuning using our available resources, we chose this approach over few-shot prompting. We applied LoRA for efficient fine-tuning, changing only a subset of the model's parameters to conserve time and computational resources, while achieving similar performance to full fine-tuning (Hu et al. 2021). The evaluation of fine-tuned FLAN-T5 models is also based on 10-fold student-level cross-validation using the same data split as was used for in-context learning with the other models. However, rather than including example explanations and their assigned labels in the prompt, they were used as training examples in a supervised learning approach. As with the in-context learning approach, we explored variants of prompts including the rubric item for each question and/or the exemplar correct response created by the instructor. A separate FLAN-T5 model was fine-tuned for each prompt variant.

6 Results

Results from all experiments are presented in Table 2. Our task involves multi-class classification, where each student response is categorized into *correct*, *partially correct*, or *incorrect*. We evaluated the explanation assessment models in terms of accuracy, macro F1, and weighted F1. As noted above, all combinations of the three different information elements (i.e., rubric, exemplar response, and student example responses) were explored for each LLM. Due to length constraints, Table 2 reports only the results of including one element at a time as well as including all types of information, while the findings from all combinations are discussed in this paper.

Across all experiments, FLAN-T5 models that were fine-tuned with rubric information and the instructor's exemplar response achieved the highest accuracy (acc.=0.824). This was a substantial improvement over the majority baseline, which always predicts the most common class (acc.=0.730), as well as the next-highest performing approach, which was GPT-4 with ten student examples included in the prompt (acc.=0.775). In terms of macro F1 score, GPT-4 with ten labeled student explanation responses included in the prompt achieved the highest performance (F1=0.664). This was a significant improvement over the majority baseline (F1=0.281) and the next-highest performing approach, which was GPT-4 with all three information elements included in the prompt (F1=0.641). In terms of weighted F1 score, FLAN-T5 models that were fine-tuned with rubric information and the instructor's exemplar response achieved the highest performance (F1=0.798). This was an improvement over the majority baseline (F1=0.616) and a small improvement over the next-highest performing approach, which was GPT-4 with ten labeled student explanation responses included in the prompt (F1=0.792).

In general, our results demonstrate that including rubric information in the prompt improved model performance. For FLAN-T5, Llama 2, and GPT-4, both accuracy and F1 score were improved relative to the prompting approach that only provided high-level instructions for the explanation assessment task. We observed the largest improvement in model performance when the sole additional information was a set of ten labeled explanation responses from other students. With this prompt, Llama 2 and GPT-4 demonstrated improved accuracy over the instruction-only approach, while Llama 2, GPT-3.5, and GPT-4 exhibited improved macro F1 scores. However, we found that including the instructor's exemplar response into the prompt led to reduced model performance across all models except for Llama 2, compared to the instruction-only approach. This reduction may stem from the exemplar responses often containing comprehensive details that exceed the question's scope, leading the models to apply a very strict standard in assessing student responses. Consequently, responses were more frequently categorized as partially correct or incorrect, even though they should be labeled correct within the question's intended scope.

Next, we looked at the effects of including two information elements in the prompt. Note that these results are omitted from Table 2 to save space. We observed that the highest accuracy and F1 score for FLAN-T5 were achieved when the models had access to both rubric information and the instructor's exemplar response. That is, we found that there was an additive effect of including multiple information elements for FLAN-T5 models. In comparison, the general trend across the prompting approaches for Llama 2, GPT-3.5, and GPT-4 that utilized two information elements was that there was not an additive benefit of including multiple information elements. For example, GPT-3.5 and GPT-4 including either rubric information or the exemplar response in addition to labeled student responses led to reduced performance compared to models that only had access to ten student example responses. In addition, Llama 2 generally demonstrated a decrease in performance when using two information elements compared to only one; however, the combination of the exemplar response and ten student responses without the rubric led to improved performance over all approaches that incorporated only one information element.

A distinct trend emerged when all three information elements were included in the prompt.

Model	Prompt Variation	Accuracy	F1 (macro)	F1 (weighted)
Majority Baseline	--	0.730	0.281	0.616
FLAN-T5-base (250M)	Fine-tuned with instructions	0.803	0.476	0.764
	Fine-tuned with instructions + Rubric	0.820	0.506	0.789
	Fine-tuned with instructions + Exemplar response	0.792	0.465	0.754
	Fine-tuned with instructions + Rubric + Exemplar response	**0.824**	0.550	**0.798**
Llama 2-7B	Instructions only	0.509	0.184	0.538
	Instructions + Rubric	0.664	0.400	0.698
	Instructions + Exemplar response	0.526	0.234	0.579
	Instructions + 10 student examples	0.706	0.443	0.717
	Instructions + Rubric + Exemplar response + 10 student examples	0.744	0.444	0.751
GPT-3.5	Instructions only	0.664	0.545	0.684
	Instructions + Rubric	0.564	0.449	0.586
	Instructions + Exemplar response	0.519	0.425	0.539
	Instructions + 10 student examples	0.612	0.591	0.644
	Instructions + Rubric + Exemplar response + 10 student examples	0.533	0.537	0.560
GPT-4	Instructions only	0.685	0.574	0.708
	Instructions + Rubric	0.709	0.606	0.732
	Instructions + Exemplar response	0.651	0.422	0.686
	Instructions + 10 student examples	0.775	**0.664**	0.792
	Instructions + Rubric + Exemplar response + 10 student examples	0.754	0.641	0.779

Table 2: Student explanation assessment results across models and prompt variations.

GPT-3.5 and GPT-4 models with access to all three information elements performed worse than models provided with only ten labeled student example responses, both in terms of accuracy and F1 scores. However, for Llama 2 models, incorporating all three information elements in the prompt resulted in the highest accuracy and F1 scores compared to any other combinations of information.

These results suggest that the best results are not necessarily guaranteed by providing the model with the maximum amount of task-related information. Models consistently performed well when the prompt included labeled examples of other students' responses, but including the instructor-created exemplar response tended to reduce model performance as discussed. Adjusting the exemplar response provided to the models, by adding clarification or simplifying its content, could potentially lead to improved performance when this information element is included. This underscores an important area for future research.

Overall, these results demonstrate that fine-tuning FLAN-T5 and utilizing few-shot learning with GPT-4 are both viable approaches to this explanation assessment task. Although FLAN-T5 requires more training data than GPT-4 to reach high performance levels (our preliminary analysis indicated that the predictive accuracy of a FLAN-T5 model, fine-tuned with only the data from five focus group students, was 60%), this tradeoff may be acceptable considering that FLAN-T5 is open-source and GPT-4 is proprietary. This consideration becomes more critical as our classroom implementation scales, especially in large classroom settings with multiple sessions where deployment costs become a significant factor. Conversely, if the EXPLAINIT system is implemented in a course where FLAN-T5 models have not been trained with student data from that course, GPT-4 with one-shot learning (with rubric information) might significantly outperform FLAN-T5, making GPT-4 potentially more suitable for the classroom response system. It will be crucial to weigh practical benefits, scalability, and cost considerations when deploying a runtime version of the explanation assessment system during the classroom use of EXPLAINIT. In practice, these results suggest that a hybrid system may be a viable approach. When a new question is deployed using the system, zero-shot learning with GPT-4 can be used based on a pre-defined rubric that was created for assessing responses to the question. Since this information can be created at the same time as the question, it can be provided to the system when the new question is first deployed. Then, as student responses to the question are collected, they can be used to fine-tune a FLAN-T5 model, which can then replace the GPT-4 model once it starts showing superior performance.

7 Conclusion

Prompting students to craft self-explanations has demonstrated to offer numerous educational advantages. However, it often requires substantial time and effort necessary for instructors to manually assess student responses and provide feedback for students, which renders them unsuitable in large classroom environments. To address this challenge, we present EXPLAINIT, a self-explanation-based classroom response system specifically designed to encourage students in formulating written self-explanations during undergraduate STEM lectures. Our NLP framework builds on Transformer-based large language models, such as FLAN-T5 and GPT-4, in assessing the correctness of student explanations, and it is evaluated using our dataset collected from classroom interactions with the EXPLAINIT system. Results demonstrate that fine-tuned FLAN-T5 models using prompts with rubric information and an exemplar response achieved the highest accuracy and weighted F1 score, while few-shot prompting that provided GPT-4 with ten labeled student response examples achieved the highest macro F1 score. These results indicate the potential to use large language models for automated explanation assessment, which can be leveraged to provide adaptive support for students' self-explanations in classroom environments.

Moving forward, there are several promising directions for future work. First, it will be important to implement the full suite of EXPLAINIT system functionalities, including NLP assessment models, in a classroom environment and investigate their impact on students' learning outcomes. It would also be interesting to incorporate AI capabilities to support question and rubric generation, thereby reducing the amount of work required by instructors to use EXPLAINIT in their classes. Additionally, the explanation assessment system could be expanded to support a finer-grained assessment of students' self-explanations. For example, concept-level assessment of students' self-explanations could provide more insightful feedback for both students and instructors. Also, it will be important to investigate this explanation assessment approach in disciplines other than computer science to evaluate its performance in other domains. Finally, it will be important to explore how different types of exemplar responses and rubric items impact model performance. If we are able to identify characteristics of exemplar responses and rubric items that most improve the predictive accuracy of our LLM-based framework for self-explanation assessment, that will enable our classroom response system to more effectively support student learning in new settings where there is limited student data that can be used to inform the assessment models.

8 Limitations

One limitation of our work is the challenge associated with evenly comparing fine-tuned models (i.e., FLAN-T5) with models that are evaluated based on few-shot in-context learning (i.e., Llama 2, GPT-3.5, and GPT-4). In our work, FLAN-T5 had access to 90% of the dataset as training data because of the 10-fold student-level cross-validation setup. In contrast, while the models that used in-context learning used the same cross-validation setup, they had access to only ten student responses that were sampled from the training set for each cross-validation fold. This limitation was a result of the practical consideration that LLMs have limited context lengths and that proprietary LLMs have monetary costs on a per-token basis. As a result, it is not feasible to provide an unlimited number of labeled student explanation responses in the prompt to an LLM, and the limit of ten student responses was chosen because it seemed reasonable. To overcome this limitation, future work could systematically investigate whether there is a more optimal number of example student responses that balances between model performance and costs. Another limitation of this work is the generalizability of the result suggesting that including an exemplar response created by the instructor in the prompt led to reduced model performance. It may be the case that certain characteristics of the exemplar responses used in this work were suboptimal for providing an LLM with guidance on how to correctly assess students' explanation responses. Further investigation into the impacts of various characteristics of exemplar responses would be helpful for addressing this limitation.

Acknowledgments

This research was supported by funding from the National Science Foundation under Grants DUE-2111473 and DUE-2111216. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Leonard Bachman and Christine Bachman. 2011. A study of classroom response system clickers: Increasing student engagement and performance in a large undergraduate lecture class on architectural research. *Journal of Interactive Learning Research*, 22(1):5-21.

John Barnett. 2006. Implementation of personal response units in very large lecture classes: Student perceptions. *Australasian Journal of Educational Technology*, 22(4):474–494. https://doi.org/10.14742/ajet.1281.

Gary Beauchamp and Steve Kennewell. 2010. Interactivity in the classroom and its impact on learning. *Computers & Education*, 54(3):759-766.

Lorena Blasco-Arcas, Isabel Buil, Blanca Hernández-Ortega, and F. Javier Sese. 2013. Using clickers in class. The role of interactivity, active collaborative learning and engagement in learning performance. *Computers & Education*, 62:102-110.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, and Dario Amodei. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877-1901.

Kirsten Butcher. 2006. Learning from text with diagrams: Promoting mental model development and inference generation. *Journal of Educational Psychology*, 98(1):182.

Jane Caldwell. 2007. Clickers in the large classroom: Current research and best-practice tips. *CBE-Life Sciences Education*, 6(1):9-20.

Julie Campbell and Richard Mayer. 2009. Questioning as an instructional method: Does it affect learning from lectures?. *Applied Cognitive Psychology*, 23(6):747-759.

Dan Carpenter, Andrew Emerson, Bradford Mott, Asmalina Saleh, Krista Glazewski, Cindy Hmelo-Silver, and James Lester. 2020. Detecting off-task behavior from student dialogue in game-based collaborative learning. In *Proceedings of the*

International Conference on Artificial Intelligence in Education.

Dan Carpenter, Michael Geden, Jonathan Rowe, Roger Azevedo, and James Lester. 2020. Automated analysis of middle school students' written reflections during game-based learning. In *Proceedings of the 21st International Conference on Artificial Intelligence in Education*.

Xinyue Chen and Xu Wang. 2022. Scaling Mixed-Methods Formative Assessments (mixFA) in Classrooms: A Clustering Pipeline to Identify Student Knowledge. In *Proceedings of the International Conference on Artificial Intelligence in Education*.

Michelene TH Chi and Ruth Wylie. 2014. The ICAP framework: Linking cognitive engagement to active learning outcomes. *Educational Psychologist*, 49(4):219-243.

Michelene TH Chi, Nicholas De Leeuw, Mei-Hung Chiu, and Christian LaVancher. 1994. Eliciting self-explanations improves understanding. *Cognitive Science*, 18(3):439-477.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, ... and Jason Wei. 2022. Scaling instruction-finetuned language models. arXiv:2210.11416.

Aubrey Condor, Max Litster, and Zachary Pardos. 2021. Automatic short answer grading with SBERT on out-of-sample questions. In *Proceedings of the International Conference for Educational Data Mining*.

Aubrey Condor, Zachary Pardos, and Marcia Linn. 2022. Representing scoring rubrics as graphs for automatic short answer grading. In *Proceedings of the International Conference on Artificial Intelligence in Education*.

Kent J. Crippen, and Boyd L. Earl. 2007. The impact of web-based worked examples and self-explanation on performance, problem solving, and self-efficacy. *Computers & Education*, 49(3):809-821.

Brenda A. Fonseca, and Michelene TH Chi. 2011. Instruction based on self-explanation. In *Handbook of Research on Learning and Instruction,* edited by R. E. Mayer and P. A. Alexander, pages 296-321. New York, NY: Routledge.

Scott Freeman, Sarah L. Eddy, Miles McDonough, Michelle K. Smith, Nnadozie Okoroafor, Hannah Jordt, and Mary Pat Wenderoth. 2014. Active learning increases student performance in science, engineering, and mathematics. In *Proceedings of the National Academy of Sciences*.

Stephanie Herppich, Jörg Wittwer, Matthias Nückles, and Alexander Renkl. 2016. Expertise amiss: interactivity fosters learning but expert tutors are less interactive than novice tutors. *Instructional Science*, 44(3):205-219.

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:2106.09685.

Nathaniel Hunsu, Olusola Adesope, and Dan Bayly. 2016. A meta-analysis of the effects of audience response systems (clicker-based technologies) on cognition and affect. *Computers & Education*, 94:102-119.

Cheryl I. Johnson and Richard E. Mayer. 2010. Applying the self-explanation principle to multimedia learning in a computer-based game-like environment. *Computers in Human Behavior*, 26(6):1246-1252.

Robin H. Kay and Ann LeSage. 2009. Examining the benefits and challenges of using audience response systems: A review of the literature. *Computers & Education*, 53(3):819-827.

Tiaoqiao Liu, Wenbiao Ding, Zhiwei Wang, Jiliang Tang, Gale Yan Huang, and Zitao Liu. 2019. Automatic short answer grading via multiway attention networks. In *Proceedings of the International Conference on Artificial Intelligence in Education*.

Jiaqi Lun, Jia Zhu, Yong Tang, and Min Yang. 2020. Multiple data augmentation strategies for improving performance on automatic short answer scoring. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Richard E. Mayer, Andrew Stull, Krista DeLeeuw, Kevin Almeroth, Bruce Bimber, Dorothy Chun, Monica Bulger, Julie Campbell, Allan Knight, and Hangjin Zhang. 2009. Clickers in college classrooms: fostering learning with questioning methods in large lecture classes. *Contemporary Educational Psychology*, 34:51–57.

Mary L. McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276-282.

Danielle S. McNamara. 2004. SERT: Self-explanation reading training. *Discourse Processes*, 38(1):1-30.

Steven Moore, Huy A. Nguyen, Norman Bier, Tanvi Domadia, and John Stamper. 2022. Assessing the quality of student-generated short answer questions using GPT-3. In *Proceedings of the European Conference on Technology Enhanced Learning*.

Bogdan Nicula, Mihai Dascalu, Tracy Arner, Renu Balyan, and Danielle S. McNamara. 2023. Automated Assessment of Comprehension Strategies from Self-Explanations Using LLMs. *Information*, 14(10), 567.

OpenAI. 2023. *GPT-4 Technical Report*.

Peter Pirolli and Margaret Recker. 1994. Learning strategies and transfer in the domain of programming. *Cognition and Instruction*, 12(3):235-275.

Raysa Rivera-Bergollo, Sami Baral, Anthony Botelho, and Neil Heffernan. 2022. Leveraging auxiliary data from similar problems to improve automatic open response scoring. In *Proceedings of the International Conference for Educational Data Mining*.

Marguerite Roy, and Michelene TH Chi. 2005. The self-explanation principle in multimedia learning. In *The Cambridge Handbook of Multimedia Learning*, edited by R.E. Mayer, pages 271-286. Cambridge University Press.

Amy M. Shapiro, Judith Sims-Knight, Grant V. O'Rielly, Paul Capaldo, Teal Pedlow, Leamarie Gordon, and Kristina Monteiro. 2017. Clickers can promote fact retention but impede conceptual understanding: The effect of the interaction between clicker use and pedagogy on learning. *Computers & Education*, 111:44-59.

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.

Andy Smith, Osman Aksit, Wookhee Min, Eric Wiebe, Bradford Mott, and James Lester. 2016. Integrating real-time drawing and writing diagnostic models: An evidence-centered design framework for multimodal science assessment. In *Proceedings of the Thirteenth International Conference on Intelligent Tutoring Systems*.

Andy Smith, Samuel Leeman-Munk, Angi Shelton, Bradford Mott, Eric Wiebe, and James Lester. 2019. A multimodal assessment framework for integrating student writing and drawing in elementary science learning. *IEEE Transactions on Learning Technologies*, 12(1):3-15.

Shunya Takano and Osamu Ichikawa. 2022. Automatic scoring of short answers using justification cues estimated by BERT. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications*.

Amirsina Torfi, Rouzbeh A. Shirvani, Yaser Keneshloo, Nader Tavaf, and Edward A. Fox. 2020. Natural language processing advancements by deep learning: A survey. *arXiv:2003.01200*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, ... and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Mengxue Zhang, Sami Baral, Neil Heffernan, and Andrew Lan. 2022. Automatic short math answer grading via in-context meta-learning. *arXiv:2205.15219*.

Harnessing GPT to Study Second Language Learner Essays: Can We Use Perplexity to Determine Linguistic Competence?

Ricardo Muñoz Sánchez[†], **Simon Dobnik**[‡], **Elena Volodina**[†]
[†] Språkbanken Text, University of Gothenburg, Sweden
[‡] CLASP, FLoV, University of Gothenburg, Sweden
{ricardo.munoz.sanchez,simon.dobnik,elena.volodina}@gu.se

Abstract

Generative language models have been used to study a wide variety of phenomena in NLP. This allows us to better understand the linguistic capabilities of those models and to better analyse the texts that we are working with. However, these studies have mainly focused on text generated by L1 speakers of English. In this paper we study whether linguistic competence of L2 learners of Swedish (through their performance on essay tasks) correlates with the perplexity of a decoder-only model (GPT-SW3). We run two sets of experiments, doing both quantitative and qualitative analyses for each of them. In the first one, we analyse the perplexities of the essays and compare them with the CEFR level of the essays, both from an essay-wide level and from a token level. In our second experiment, we compare the perplexity of an L2 learner essay with a normalised version of it. We find that the perplexity of essays tends to be lower for higher CEFR levels and that normalised essays have a lower perplexity than the original versions. Moreover, we find that different factors can lead to spikes in perplexity, not all of them being related to L2 learner language.

1 Introduction

In the past couple of years we have seen a fast development in the capabilities of decoder-only language models, such as GPT-4 (OpenAI et al., 2024), LLaMA (Touvron et al., 2023), and BLOOM.[1] These models have been increasingly deployed in a wide variety of applications such as machine translation (Qian, 2023) and financial (Li et al., 2023) and legal applications (Kwak et al., 2023). In the context of second language (L2) educational applications, these models have been deployed to different subtasks, with varying degrees of success (Naismith et al., 2023; Yancey et al., 2023).

Even though they excel at a multitude of NLP tasks, the inner workings of large language models are obscure. This means that it is complicated, if not impossible, to verify that a model has learned actual linguistic features instead of making spurious correlations. The same issue is true when attempting to determine how the model arrived at specific decisions (Blevins et al., 2023). This can be tricky, especially in high-stakes situations such as educational applications.

One such example is the evaluation and assessment of second language performance, as the result of such assessment can alter the life opportunities a person has access to (education, job offers, among others). When dealing with text, we want to be able to understand how systems interact with text from second language learners. This would allow us to properly build models for tasks such as second language assessment, for grammatical error correction, among others, while complying with the right to an explanation (Official Journal, 2016).

One way to do so is to analyse how much text diverges from what a language model expects. This can be done using perplexity, a statistical concept that measures the probability of a sequence given an estimator and has been interpreted in an intuitive manner as a measure of "surprisal" (Dobnik et al., 2018; Niu and Penn, 2020). However, it has been mostly used to study texts of first language speakers of English. To address this gap, we aim to study how perplexity interacts with texts generated by second language learners of Swedish.

We hypothesise that the perplexity of a decoder-only model is related to the complexity of the text in an L2 speaker's essay. In this paper we aim to answer the following research questions:

- To what degree can the linguistic competence of a learner (as evidenced in an essay) be estimated using perplexity from language models?

[1] https://bigscience.notion.site/BLOOM-BigScience-176B-Model-ad073ca07cdf479398d5f95d88e218c4

June 20, 2024 ©2024 Association for Computational Linguistics

- To what degree does the perplexity of the language model correspond to CEFR[2] levels?

We have used GPT-SW3 (Ekgren et al., 2023) as our language model for this study. It is based on the GPT series (Radford et al., 2019; Brown et al., 2020) and trained on data of several Nordic languages. We give more details about it in Section 3.1. For the L2 learner essays we used Swell (Volodina et al., 2016a; Volodina, 2024), a collection of corpora of L2 learner essays in Swedish. A more in-depth explanation of its contents and how the essays were collected can be found in Section 3.2. We describe perplexity and some of its intuitive interpretations in Section 3.3.

We ran two sets of experiments. In Section 4 we show the perplexity of the essays and see how it is distributed both across levels and within the essays. This analysis is done both in a statistical manner in Section 4.1 and in a linguistic manner in Section 4.2. The second set of experiments is a comparison between the perplexity from original essays written by L2 students and normalised versions of these essays, described in Section 5.

2 Related Work

2.1 NLP for Second Language Learning

There have been several ways in which NLP has been used for second language learning. The two most relevant for us are automated essay scoring and grammatical error correction.

Automated essay scoring (AES) of L2 learner texts is a task in which we have a system that takes a text generated by an L2 learner and assigns a grade or level to it. This can be done using CEFR levels, but different levels or scales have also been used in the past. Despite their ubiquity in NLP and machine learning in general, deep learning had not been used in AES until 2016 (Alikaniotis et al., 2016; Taghipour and Ng, 2016). Even though there have been more works that use deep learning for this task, Mayfield and Black (2020) warn that their performance might not be good enough to justify the lack of transparency and the increased computational costs. As for decoder-only models, they were first used for this task in 2023, with mixed results (Naismith et al., 2023; Yancey et al., 2023).

Grammatical error correction (GEC) is a task in which we have a system that takes a text assumed to have some sort of error or non-standard language and returns a normalised version of the same text. It is important to note that, despite the name of the task, the errors in the original (or source) text are not limited to grammar and often include other kinds or errors, such as lexical, orthographic, among others (Bryant et al., 2017). It is often seen as variation of machine translation, with the source language being the non-normalised language and the target being the normalised one (Wang et al., 2021). Because of this, sequence-to-sequence neural models have often been used for this task, including decoder-only models (e.g. Flachs et al., 2019).

Most of the work done so far in this area has focused on English. However, the advances for Swedish are scarce, despite it being a language with relatively good language technology resources. The Swell corpus collection (Volodina, 2024) contains corpora both for AES (Swell-pilot) and for GEC (Swell-gold). As far as we are aware, the current state of the art of AES in Swedish is that of Pilán et al. (2016) and Volodina et al. (2016b). They use a feature-based approach using length-based, lexical, morphological, syntactic, and semantic features. In terms of GEC, the most recent approach is that by Kurfalı and Östling (2023), who used a transformer-based model.

2.2 Language models as Predictors of Grammaticality

As Lappin (2023) points out, the discussion of linguistic capabilities of large language models ranges from (overstated) claims of their sentience and the arrival of artificial general intelligence to skepticism and dismissal. Because of that, he argues it is is essential to explore the capabilities of these models. One way that this has been done is by evaluating how much texts generated by humans diverge from what a language model expects.

One possible approach is by evaluating the grammaticality or linguistic acceptability of a text. The idea is to give a system a text that it has to determine whether it is grammatically correct or not. There are two main approaches through which this can be done. The first one is as a classification task, assigning each sentence a class that determines whether a sentence is grammatically correct or not (Klezl et al., 2022).

Another approach is by checking whether a text

[2] CEFR stands for Common European Framework of Reference for Languages. It is a framework to evaluate foreign language learning and assigns one of six reference levels to determine the proficiency level of a second language speaker (Council of Europe, 2001).

is likely to appear in text generated by a language model or not (Lau et al., 2017). In particular, perplexity has been used as a way to determine how much a model expects the tokens within a text to appear (Niu and Penn, 2020). It has subsequently been interpreted as a measure of "surprisal".

3 Methodology and Experimental Settings

3.1 GPT-SW3

Our objective is to determine how much L2 Swedish learner's texts differ from what a generative language model would expect.

In order to do this, we use GPT-SW3 (Ekgren et al., 2023), an auto-regressive model based on the GPT series of models (Radford et al., 2019; Brown et al., 2020). It was trained on a large dataset called The Nordic Pile (Öhman et al., 2023), a 1.3TB dataset containing large dumps of several websites in the Nordic languages.[3]

We decided to use this model as it is to our knowledge the largest and best performing generative model currently available for the Swedish language. Our assumption is that it will be able to model Swedish in a similar way to how L1 speakers write across a variety of domains, thus allowing it to identify when an L2 speaker's sentences differ from what an L1 speaker would write.

3.2 Dataset

To compare how GPT-SW3 works for different CEFR levels of language learner essays, we have used the Swell corpus collection (Volodina, 2024). It is divided into two corpora, Swell-pilot (Volodina et al., 2016a) and Swell-gold (Volodina et al., 2019).

Swell-pilot consists of 502 essays divided into three sub-corpora, collected between 2012 and 2016. All essays are anonymised and annotated for CEFR level. However, there are six essays that lack a level, so we have ignored them for the purposes of this paper.

Swell-gold consists of 502 essays that were collected between 2017 and 2021. They are pseudonymised and include both the original version and a normalised version of the essays. They also contain level indications, which, however, do not align with the CEFR levels. These levels were determined by using the course that the student was taking as a proxy for proficiency, not by an actual analysis of learner performance.

Level	N° of Essays
A1	59
A2	143
B1	86
B2	105
C1	96
C2	7 (Swell-pilot) 43 (Swell-gold)

Table 1: Distribution of the CEFR levels in Swell-pilot. Note that we added extra essays from Swell-gold to have a more representative sample of the C2 level.

Level	N° of Essays
Beginner (*Nybörjare*)	289
Intermediate (*Fortsättning*)	45
Advanced (*Avancerad*)	168

Table 2: Distribution of the proficiency levels in Swell-gold. The text in parenthesis is the name for the level used in the metadata (in Swedish).

In our first experiment (Section 4) we use all the essays from Swell-pilot that have a CEFR level assigned to them, as they showcase a good distribution from the different levels, as seen from Table 1. The only exception is the C2 level which only has seven essays. To make up for this, we randomly sampled 43 of the normalised version of the essays in Swell-gold by advanced speakers as we assumed that it would more closely resemble those by C2 second language learners.

In our second experiment (see Section 5) we use both the original and the normalised versions of all of the Swell-gold essays. The distributions of the proficiency annotations of the essays can be found in Table 2.

3.3 Perplexity as a Measure of Surprisal

Perplexity is the probability that an observation is made by an estimator. When dealing with generative models, this is the probability that a sequence S appears in a natural language L. When we use a language model M, we do so as it approximates (or models) language L. Thus, we can intuitively interpret the perplexity PP_M as a way to measure how "surprised" the model M is to see sequence S.

[3] The languages included are Danish, Faroese, Icelandic, Norwegian, and Swedish. For more information about the contents of the dataset, read AI Sweden's blog post: https://medium.com/ai-sweden/the-nordic-pile-a8d5aaf3db60

Now, perplexity is defined in mathematical terms as the probability that an estimator (in this case M) sees an observation S (Jelinek et al., 2005). The best way to calculate this for a generative model is by taking the product of the probabilities of a token given the previously generated ones:

$$PP_M(S) = \mathbb{P}(S)^{-|S|} = \prod_{i \leq |S|} \mathbb{P}(S_i|S_{<i})^{-|S|}$$

where S_i denotes the i-th token of S and $S_{<i}$ the sequence S up to S_i.

Given that this is a very small number, we risk having an underflow in our calculations[4]. Because of this, we are better off using the log-likelihood as opposed to using the regular likelihood. Thus, we have

$$\begin{aligned}\log PP_M(S) &= \log \prod_{i \leq |S|} \mathbb{P}(S_i|S_{<i})^{-|S|} \\ &= -\frac{1}{|S|} \sum_{i \leq |S|} \mathbb{P}(S_i|S_{<i})\end{aligned}$$

On the other hand, cross-entropy is a way to measure how much the information between two probability distributions differs. It is often used as the loss function for classification tasks in machine learning (Song et al., 2023), including text generation. When one of the distributions is unknown (as is the case when dealing with language modeling), it can be estimated as follows:

$$\mathcal{C}(S) = Loss_M(S) = -\frac{1}{|S|} \sum_{i \leq |S|} \mathbb{P}(S_i|S_{<i})$$

Thus, we can calculate the perplexity for S as the mean cross-entropy for S given a generative model M:

$$\log PP_M(S) = Loss_M(S)$$

Moreover, given that the relation between likelihood and cross-entropy is given by a monotonic function, the relative positions between different data points does not change. This means that we can use the loss from GPT-SW3 (M in this case) to determine the perplexity of a given essay (S in this case). For the rest of this paper we will refer by perplexity to $-\log PP_M(S)$ as opposed to $PP_M(S)$. This is due to the fact that the second number is more likely to underflow as it is a multiplication of probabilities.

Level	Mean	Median	Std
A1	5.25	5.01	0.78
A2	4.49	4.49	0.74
B1	4.13	4.15	0.48
B2	3.96	3.95	0.42
C1	3.67	3.60	0.36
C2	3.46	3.48	0.48

Table 3: Statistics on the perplexities of GPT-SW3 on the Swell-pilot essays per level. Note that all values get lower the more advanced the students are. This is an indication that as L2 learners advance in their journey, their language approaches that of the language model, which we are assuming should be close to that of an L1 speaker.

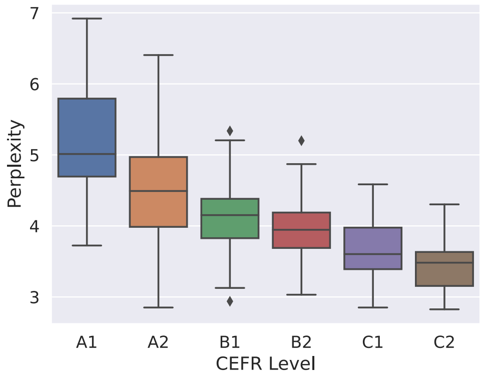


Figure 1: Boxplots for the perplexities of the different CEFR levels. As we can see, as the L2 learner's level progresses, the perplexity of their texts according to GPT-SW3 diminishes.

4 Experiment 1: Perplexity and CEFR Levels

In this section we analyse the perplexities of the essays given by GPT-SW3. We begin by doing statistical analyses of the perplexities by level in Section 4.1. We then do a linguistic analysis of some of the essays of each level in Section 4.2 with the aim of identifying patterns in how the perplexities are distributed within the essay texts.

4.1 Quantitative Analysis

As we can see from Figure 1 and Table 3, the essays from more advanced learners tend to have lower perplexity than those of less advanced learners. This is evidenced when looking both at the mean and the median values of the perplexities for each CEFR level as the more advanced levels have a lower mean and median value.

[4] An underflow occurs when small numbers are rounded down to zero by the computer due to how floating-point numbers work.

When looking at the boxplots in Figure 1 we see a similar pattern appear. For each subsequent level, the boundaries of the same quartile are noticeably lower than of the previous level. For example, the first quartile's roof and floor values in level A1 are higher from the ones in level A2, which are in turn higher from the ones in level B1, and so on. The exceptions to this are the boxplots of levels C1 and C2, which have somewhat similar distributions. A possible explanation for this could be that both of these levels are considered to be essentially fluent in the target language, meaning that both kinds of L2 speakers would be able to produce pretty similar sentences. Another possible explanation could be that the normalized essays chosen as a substitute for C2 level essays have a higher perplexity than actual C2 level essays.

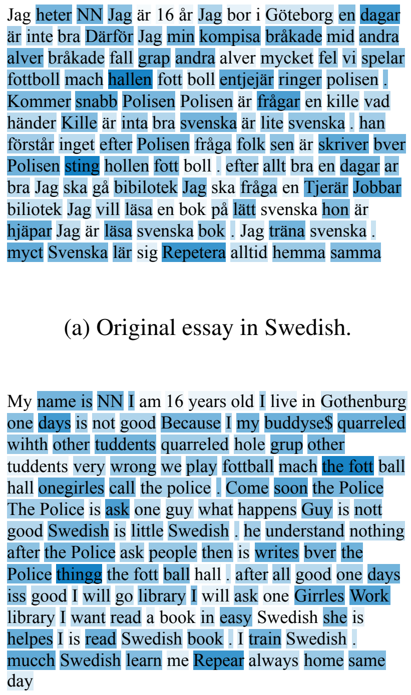
However, it is also important to note that the boxplots still have a big overlap between levels, especially in adjacent ones. This means that, while there is a tendency for the perplexities of the essays from GPT-SW3 to get lower the more advanced a learner is, it is by no means a strong indicator for determining the CEFR level of a Swedish L2 learner. This makes sense as language learning itself is a continuous endeavor, as opposed to a discrete one (e.g., Ortega, 2012).

Finally, when looking at the standard deviations, we can see that they also get lower the higher the level. A possible explanation for this could be that the more advanced a learner is, the more likely they are to experiment with the language within certain boundaries that they know to work.

4.2 Qualitative Analysis

To better understand the phenomenon of perplexity, we have also carried out a qualitative analysis. We have selected essays for this qualitative analysis in the following way: we ignored the essays that were outliers in terms of perplexity on each level; of the remaining essays, we picked the two with the highest perplexity, the two with the lowest, and the two closest to the median in their respective levels; and level C2 was ignored from this analysis due to its similarity to the C1 essays. This leaves us with six essays per level for a total of 30 essays.

The analysis has several aspects that we have chosen to focus on. First of all, we want to see what the value of the perplexity depends upon in linguistic terms when seen on a token level and whether this correlates with the CEFR levels. We also want to know whether the perplexities within an essay can be used to help guide or inform on possible aspects on which to focus when grading an essay. Finally, we want to understand how perplexity in LLMs works when dealing with text that was generated by an L2 language learner.



Jag heter NN Jag är 16 år Jag bor i Göteborg en dagar är inte bra Därför Jag min kompisa bråkade mid andra alver bråkade fall grap andra alver mycket fel vi spelar fottboll mach hallen fott boll entjejär ringer polisen . Kommer snabb Polisen Polisen är frågar en kille vad händer Kille är inta bra svenska är lite svenska . han förstår inget efter Polisen fråga folk sen är skriver bver Polisen sting hollen fott boll . efter allt bra en dagar ar bra Jag ska gå bibilotek Jag ska fråga en Tjerär Jobbar biliotek Jag vill läsa en bok på lätt svenska hon är hjäpar Jag är läsa svenska bok . Jag träna svenska . myct Svenska lär sig Repetera alltid hemma samma

(a) Original essay in Swedish.

My name is NN I am 16 years old I live in Gothenburg one days is not good Because I my buddyse$ quarreled wihth other tuddents quarreled hole grup other tuddents very wrong we play fottball mach the fott ball hall onegirles call the police . Come soon the Police The Police is ask one guy what happens Guy is nott good Swedish is little Swedish . he understand nothing after the Police ask people then is writes bver the Police thingg the fott ball hall . after all good one days iss good I will go library I will ask one Girrles Work library I want read a book in easy Swedish she is helpes I is read Swedish book . I train Swedish . mucch Swedish learn me Repear always home same day

(b) Translated version of the essay to English.

Figure 2: An example of an A1 level with median perplexity. Darker colors correspond with higher perplexities. Note that the translation was made with the aim of the text being aligned while trying to replicate grammatical errors and misspellings found in the original text.

In more practical terms, our intention is to examine whether variations in perplexity can be explained by the linguistic competence of a learner. We focus particularly on sections and tokens with high perplexity, setting a threshold at 6 based on the analysis of graphs in Appendix B, showing how perplexity is distributed across the essays and their variation across different levels. We then analyse the tokens above this threshold across several dimensions.

4.2.1 Placement Within an Essay

The first hypothesis we have explored is that tokens at the beginning of an essay would have higher perplexity values. The idea is that the first few words would be relatively more unexpected for the language model than the text found later in the essay. Looking through the different levels, we can state that this is indeed true in most cases, as can be seen in Appendix A.

This is more noticeable at lower levels, especially if an essay starts with the pronoun *jag*[5] and its various forms. Two examples of this are *Jag heter NN [...]*[6] or **Min skolan ligger [...]*[7]. The lowest perplexity for the first tokens in an essay can be observed in essays starting with the formal subject *Det är / Det finns*[8].

The high perplexity at the beginning of an essay does not seem to characterize essays of a certain level. Therefore, it could be reasonable to ignore the perplexities of the first five or six tokens for successful practical applications of perplexity for L2 essays. Another option would be to weight the perplexity scores depending on their position in the essay.

We also observe that the perplexity values tend, in general, to go down by the end of an essay. This could be because the model knows better what to expect due to the preceding context. Exceptions would arise where other unexpected elements, such as errors, may occur by the end of an essay.

4.2.2 Placement Within a Sentence

The second hypothesis we have explored is that tokens at the beginning of a sentence would have higher perplexity values. It has proven not to be the case.

Essays at levels A1 and A2 can exhibit lack of end-of-sentence punctuation, which makes it next to impossible to separate the increase in perplexity due to the beginning of a sentence with the increase in perplexity due to having a run-on sentence. Essays at levels B1, B2 and C1 do not show regular spikes in perplexity at the beginning of individual sentences. Where such spikes have been observed, this was due to other linguistic reasons, such as errors, rare words, some subjunctions, register switch (from formal to informal or vice versa) or contextually unexpected turn in narration.

Based on this analysis we suggest that the perplexity spikes at the beginning of a sentence could be treated as any other within an essay. This is supported by the fact that GPT-SW3 looks at strings of tokens, which tend to be longer than sentences.

4.2.3 Parts of Speech

Another hypothesis we have explored is that different parts of speech would have different perplexity values in general. The distribution of parts of speech among tokens with higher perplexity shows that content words[9] are much more often perplexing for the model. The percentage of content words of high perplexity is about ∼55-70%. Meanwhile, only ∼15-20% of all the words with high perplexity are function words.[10] The rest of the words with high perplexity are constituted by proper names, numerals, modal verbs, and punctuation.

The high representation of content words suggests a strong impact of semantics, contextual use, and fixed expressions on the probabilities of words expected to be used. A large number of the content words with high perplexity can be explained by various errors, such as non-idiomatic usage, incorrect spelling, or morphological errors. An example of non-idiomatic usage would be *efter allt*, which would be translated word-for-word to English as *after all*. However, this expression is not used in Swedish.

On the other hand, a high perplexity in function words is more often than not triggered by syntax errors, such as missing words or punctuation, issues in word order, and to a lesser degree by misspellings.

An interesting case is presented by high perplexities in multi-word expressions (MWEs). Quite a few of those combine with rare words that appear in combination with just a few other words. Our model is therefore triggered to expect a certain word once the initial part of an MWE is used, such as *å* in the expression *å ena sida..., å andra sidan*.[11] When, the form **å annan sidan* is used instead, the system flags the word *annan*[12] as a perplexing one. In another case, the initial preposition was omitted by a learner from the expression *i alla fall*,[13] so the system flagged *alla* as highly perplexing, whereas the error depended on the omitted token *i*. This same concept can be seen with phrases. That is, perplexity tends to be lower within, say, a noun phrase as words inside it become more predictable.

The last comment on the effects of parts of

[5] This pronoun is the equivalent of the pronoun *I* in English.

[6] Can be translated to English as *My name is NN [...]*.

[7] Can be translated to English as **My the school lies [...]*. Note the use of non-standard language by using a possessive and a determinant on a noun.

[8] Can be translated to English as *There is / There exists*.

[9] Nouns, verbs, adjectives, and adverbs.

[10] Such as prepositions, articles, particles, conjunctions, and pronouns.

[11] This can be translated to English as *on the one hand..., on the other hand*.

[12] The indefinite form of the word *other*.

[13] Can be translated to English as *anyway*.

speech on perplexity needs to be made in connection to proper nouns and names. Names are highly perplexing in general in our data, but even more interesting is the fact that some are significantly more so than others. For example, perplexity for *Kanada* [14] is much higher than for *Afrika* [15]. While we do not have enough proper names in the 30 essays we have selected for qualitative analysis to make any generalisations, we consider that this is a direction that is worth pursuing, especially in relation to possible demographic biases in data and models.

4.2.4 Punctuation

Regarding punctuation, we did not originally expect it to factor significantly into the perplexity of the text. However, we found that the highest spikes have been observed in the use of citation marks and apostrophes. Apostrophes are not used in standard Swedish, which can explain its effect on the model. Meanwhile, the perplexity spikes caused by citation marks could be explained by their low use in the training data for our model. Since citation marks are used at higher proficiency levels (at least in the Swell-pilot data), their high perplexity values may effect the assignment of a CEFR level.

As a take-away lesson, we consider that punctuation in general adds noise and should be exempt from perplexity calculations in connection to essay grading.

4.2.5 Errors

Spikes in perplexities in the running essay text show relatively strong correlation with errors. The majority of words with high perplexity contain some kind of error, either on the token itself (misspelling, morphology, etc) or errors within the previous context (word order, missing punctuation or missing syntactic word, etc).

About ∼65-80% of the highly perplexing tokens in essays at A1 and A2 level are related to errors of various types. This number gradually decreases up to the point where at B2 level and higher less than 50% of high perplexity words have a straightforward error associated. In some cases high perplexity may be explained through a (rather vague) notion of non-idiomatic language, use of relatively rare words, deponent verb forms[16], creative compounding, register, abbreviation, etc. The analysis even suggests that word tautology is punished by perplexity, i.e. an overuse of the same content word in close context.

We can summarise this by saying that in the majority of cases, high perplexities reflect an error on the token, or on the previous token. Spelling, morphology, and to a lesser degree syntax are the main reasons of high perplexity in the running text. Wrong word choice, informal register of a word, and non-idiomatic or semantically inappropriate words are also among the errors that can explain higher perplexities in our model.

However, error prediction based on perplexity, is not straightforward, since the high perplexity of a correctly used token may depend on an erroneous usage of the token before. Moreover, errors are not systematically causing high perplexity scores. At lower levels words exhibiting errors with misspellings, capitalisation, morphology and missing punctuation might receive relatively low perplexities. This apparent lack of systematicity could be explained through some of the effects that we have seen in other sections of this analysis, such as lexical choice and frequency effects, the location of the error within the text, among others.

4.2.6 Frequency effects

Given that perplexity is based on probability distributions of the tokens, it is dependent on the frequency of tokens in the dataset on which the language model was trained on.

While we noticed that frequency of vocabulary has a strong correlation with perplexities, a more systematic analysis of word frequencies against perplexity of words in sentence is left for future work.

One of the things that we noticed is that while rare words tend to have higher perplexity values, frequent words like conjunction *och*,[17] the personal pronoun *jag*,[18] and the link verb *att vara*[19] have varying perplexities, depending on their context and neighbouring words.

Another interesting observation with regards to frequency are formulaic expressions that go through language variation. For example, *kommer att* is an expression that can indicate something about the future. A second way to write this would be to drop the *att* particle. However, this second use is not widely spread and is reflected more sparsely

[14] *Canada*

[15] *Africa*

[16] E.g. *bildades*, translated to English as *were built*.

[17] Equivalent to *and* in English.

[18] Equivalent to *I* in English.

[19] Which can be translated to English as the verb *to be*.

Level	Mean	Median	Std
Beginner	4.13	4.09	0.58
Intermediate	4.28	4.32	0.42
Advanced	3.59	3.55	0.45

Table 4: Statistics on the perplexities of GPT-SWE3 on the original Swell-gold essays per level. Even though the beginner-level essays have lower mean and median values when compared to the intermediate-level essays, they have a higher standard deviation.

in online data (Berdicevskis et al., 2024). Our data analysis shows that in cases where *att* has been dropped, the content verb coming after *kommer* gets high perplexity score, as if the model expects *att* but sees a verb instead. In cases where *att* is preserved, the perplexity is on the low level on all tokens.

5 Experiment 2: Perplexity and Text Normalisation

In this section we analyse whether the perplexity of an essay given by GTP-SW3 is reduced when dealing with a normalised version of it. The idea is to establish whether non-standard language correlates with perplexity and to what degree.

When looking at the perplexities in the Swell-gold dataset in Figure 3 and Table 4, we notice that there is not a clear pattern regarding the proficiency level. While this appears to contradict the findings of Section 4.1, this could be due to how the labels were obtained. As mentioned in Section 3.2, these labels were assigned according to the course students are taking, as opposed to actual learner performance.

When comparing the original and the normalised versions of the essays, we see two noticeable tendencies. The first is that, even though the original essays seem to have different distributions depending on their level, the normalised ones have pretty much the same distribution regardless of it, as seen in Tables 4 and 5.

The other tendency is that the perplexity between the original and the normalised versions of the essays go down in all of them. Even though we have suspected this when first looking at Figure 3, the fact that there is an overlap in the boxplots should not be ignored. However, this is confirmed when looking at the figures in Table 6. Here we notice that the minimum difference between the perplexity of the original essays and their normalised versions is still positive, confirming our hypothesis that the perplexity of an essay goes down after its normalisation.

Level	Mean	Median	Std
Beginner	3.05	3.02	0.28
Intermediate	3.11	3.11	0.26
Advanced	3.10	3.08	0.27

Table 5: Statistics on the perplexities of GPT-SWE3 on the normalised Swell-gold essays per level. Note that all of the statistics from these essays are much more closer to each other across levels when compared to those of the original essays (Table 4).

Level	Mean	Median	Std	Min
Beginner	1.07	1.03	0.48	0.13
Intermediate	1.16	1.12	0.31	0.52
Advanced	0.49	0.43	0.30	0.05

Table 6: Statistics on the difference between the perplexities of GPT-SWE3 on the original and the normalised Swell-gold essays per level. Note that the minimum values of the difference are all positive, meaning that the perplexity of the normalised essays is always lower than that of their respective original essay.

These results corroborate the findings from Section 4.2. That is, the biggest effect of learner language on perplexity comes from errors and the use of non-normative language. This confirms our hypothesis that perplexity is indicative of learner language at different levels.

The remaining spikes in perplexity in normalized essays indicate use of rare words, potentially register switches, citation marks, among others.

6 Conclusions

One of the issues with large language models tends to be their lack of interpretability and explainability. This keeps true with generative models such as those based on the GPT architecture despite them being able to generate text "justifying" their reasoning (Blevins et al., 2023).

In this work we aimed to explore the relationship between the perplexity from a decoder-only model of Swedish and the complexity of the text of an L2 speaker's essays.

We found that there is an inverse relationship between the CEFR level of an essay and its perplexity. However, due to the overlap between the values of each level means that they are not a strong indica-

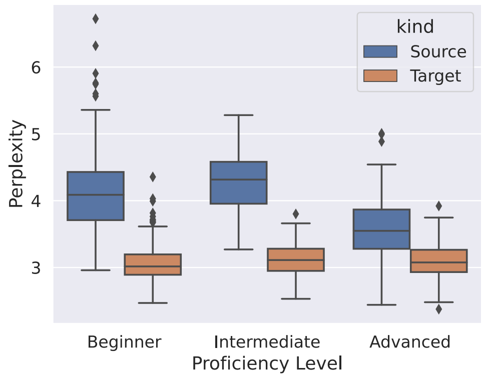


Figure 3: Boxplots for the perplexities of the different proficiency levels. Even though there does not seem to be an obvious pattern between the levels of the texts and their perplexities, the normalised texts show a much lower perplexity than the original texts. Moreover, the distributions of the normalised texts are much more similar to each other than to their original versions.

tor for the level of the essay. Moreover, we found evidence that proficiency levels derived from the course a student is taking might not be indicative of the actual proficiency of the essays.

We also found that there are perplexity effects through the essays that are not exclusive to L2 language, such as placement within a text, punctuation, frequency of the tokens, among others. Despite that, some of the more prevalent effects are characteristic of L2 language, such as errors, non-idiomatic use of the language, and multi-word expressions.

There is a correlation between the use of non-standard language as an L2 language learner. This conclusion can be drawn by the fact that the perplexity for every essay became lower after being normalised.

One of the possible applications of these could be done through the use of these features to help guide human graders with the assessment of learner language. The idea of this is to take either a human-in-the-loop (Wu et al., 2022) or a hybrid intelligence approach to evaluation (Dellermann et al., 2019). However, it would be of essence to disentangle the perplexity effects that are specific to L2 speakers from those effects that are not. This would allow us to have a more reliable and fair estimation. This, however, remains to be explored in the future.

CEFR are categorical classes used to describe language proficiency for teaching and assessment convenience. Despite that, language development itself works as a continuum, where essays within each particular level are not homogeneous with regards to their linguistic complexity. This continuum of linguistic complexity of learner language has rather vague and arbitrary cut-off points between one level and another (Hulstijn et al., 2010; Ortega, 2012; Alfter et al., 2021). Given the context of our experiments, we hypothesise that the perplexity score per essay can help place each essay on a scale between one level and another and that it may be an indirect way of grading essays within the same level. However, this is a hypothesis that needs to be explored in another paper.

7 Limitations

Throughout this paper we have talked about perplexity as a way to measure the surprisal of a model. While this is a useful way to interpret this value in an intuitive manner, it is important to note that this is just a metaphor. We are not treating the language model as an agend and humanising it. This is particularly relevant as they still have a vast amount of limitations and their misuse can lead to undesirable results (Weidinger et al., 2022).

8 Ethical Considerations

In high stakes situations such as those related to language learning it is important to constantly audit our systems and processes to ensure that unfairness does not begin to creep into the process. Moreover, we consider that a human-in-the-loop approach is the correct way to go about, as mentioned in Section 6. This allows the students to ask both for explanations on the results and for a revision of these in case they consider them to be erroneous.

Acknowledgements

The research presented here has been enabled by the Swedish national research infrastructure Nationella språkbanken, funded jointly by the Swedish Research Council (2018–2024, contract 2017-00626) and the 10 participating partner institutions.

This work is part of a continuous endeavour on exploring the use of machine learning techniques to study second language learner texts in Swedish. Moreover, we aim to modernise the L2 tools that are currently available through the research infrastructure of Språkbanken Text, in particular via Språkbanken's learning platform Lärka,[20] which

[20] https://spraakbanken.gu.se/larka/

allows researchers, as well as teachers and learners, to interact and analyse these kinds of texts in an automated manner.

The second author is also supported by a grant from the Swedish Research Council (VR project 2014-39) for the establishment of the *Centre for Linguistic Theory and Studies in Probability (CLASP)* at the University of Gothenburg.

References

David Alfter, Therese Lindström Tiedemann, and Elena Volodina. 2021. Crowdsourcing relative rankings of multi-word expressions: Experts versus non-experts. *The Northern European Journal of Language Technology (NEJLT)*, Vol.7 No.1:1–35.

Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. 2016. Automatic text scoring using neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 715–725, Berlin, Germany. Association for Computational Linguistics.

Aleksandrs Berdicevskis, Evie Coussé, Alexander Koplenig, and Yvonne Adesam. 2024. To drop or not to drop? predicting the omission of the infinitival marker in a swedish future construction. *Corpus Linguistics and Linguistic Theory*, 20(1):219–261.

Terra Blevins, Hila Gonen, and Luke Zettlemoyer. 2023. Prompting language models for linguistic structure. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6649–6663, Toronto, Canada. 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. *arXiv preprint*, arXiv:2005.14165.

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.

COE Council of Europe. 2001. *Common European Framework of Reference for languages: Learning, teaching, assessment*. Cambridge University Press.

Dominik Dellermann, Philipp Ebel, Matthias Söllner, and Jan Marco Leimeister. 2019. Hybrid intelligence. *Business & Information Systems Engineering*, 61(5):637–643.

Simon Dobnik, Mehdi Ghanimifard, and John Kelleher. 2018. Exploring the functional and geometric bias of spatial relations using neural language models. In *Proceedings of the First International Workshop on Spatial Language Understanding*, pages 1–11, New Orleans. Association for Computational Linguistics.

Ariel Ekgren, Amaru Cuba Gyllensten, Felix Stollenwerk, Joey Öhman, Tim Isbister, Evangelia Gogoulou, Fredrik Carlsson, Alice Heiman, Judit Casademont, and Magnus Sahlgren. 2023. GPT-SW3: An autoregressive language model for the nordic languages. *arXiv preprint*, arXiv:2305.12987.

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.

Jan H Hulstijn, J Charles Alderson, and Rob Schoonen. 2010. Developmental stages in second-language acquisition and levels of second-language proficiency: Are there links between them. *Communicative proficiency and linguistic development: Intersections between SLA and language testing research*, pages 11–20.

F. Jelinek, R. L. Mercer, L. R. Bahl, and J. K. Baker. 2005. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63.

Julia Klezl, Yousuf Ali Mohammed, and Elena Volodina. 2022. Exploring linguistic acceptability in Swedish learners' language. In *Proceedings of the 11th Workshop on NLP for Computer Assisted Language Learning*, pages 84–94, Louvain-la-Neuve, Belgium. LiU Electronic Press.

Murathan Kurfalı and Robert Östling. 2023. A distantly supervised grammatical error detection/correction system for swedish. *Swedish Language Technology Conference and NLP4CALL*, pages 35–39.

Alice Kwak, Cheonkam Jeong, Gaetano Forte, Derek Bambauer, Clayton Morrison, and Mihai Surdeanu. 2023. Information extraction from legal wills: How well does GPT-4 do? In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4336–4353, Singapore. Association for Computational Linguistics.

Shalom Lappin. 2023. Assessing the strengths and weaknesses of large language models. *Journal of Logic, Language and Information*, 33(1):9–20.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2017. Grammaticality, acceptability, and probability: A probabilistic view of linguistic knowledge. *Cognitive Science*, 41(5):1202–1241.

Xianzhi Li, Samuel Chan, Xiaodan Zhu, Yulong Pei, Zhiqiang Ma, Xiaomo Liu, and Sameena Shah. 2023. Are ChatGPT and GPT-4 general-purpose solvers for financial text analytics? a study on several typical tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 408–422, Singapore. Association for Computational Linguistics.

Elijah Mayfield and Alan W Black. 2020. Should you fine-tune BERT for automated essay scoring? In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 151–162, Seattle, WA, USA → Online. Association for Computational Linguistics.

Ben Naismith, Phoebe Mulcaire, and Jill Burstein. 2023. Automated evaluation of written discourse coherence using GPT-4. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 394–403, Toronto, Canada. Association for Computational Linguistics.

Jingcheng Niu and Gerald Penn. 2020. Grammaticality and language modelling. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 110–119, Online. Association for Computational Linguistics.

Official Journal. 2016. Recital 71. *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance)*, L 119.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, C. J. Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. GPT-4 technical report. *arXiv preprint*, arXiv:2303.08774.

Lourdes Ortega. 2012. Interlanguage complexity. *Linguistic complexity: Second language acquisition, indigenization, contact*, 13:127.

Ildikó Pilán, Elena Volodina, and Torsten Zesch. 2016. Predicting proficiency levels in learner writings by transferring a linguistic complexity model from expert-written coursebooks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2101–2111, Osaka, Japan. The COLING 2016 Organizing Committee.

Ming Qian. 2023. Performance evaluation on human-machine teaming augmented machine translation enabled by GPT-4. In *Proceedings of the First Workshop on NLP Tools and Resources for Translation and Interpreting Applications*, pages 20–31, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. *Language Models are Unsupervised Multitask Learners*. White Paper. Open AI.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2023. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):8135–8153.

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.

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. *arXiv preprint*, arXiv:2302.13971.

Elena Volodina. 2024. On two SweLL learner corpora – SweLL-pilot and SweLL-gold. *Huminfra Conference*, pages 83–94.

Elena Volodina, Lena Granstedt, Arild Matsson, Beáta Megyesi, Ildikó Pilán, Julia Prentice, Dan Rosén, Lisa Rudebeck, Carl-Johan Schenström, Gunlög Sundberg, and Mats Wirén. 2019. The SweLL language learner corpus: From design to annotation. *Northern European Journal of Language Technology*, 6:67–104.

Elena Volodina, Ildikó Pilán, Ingegerd Enström, Lorena Llozhi, Peter Lundkvist, Gunlög Sundberg, and Monica Sandell. 2016a. SweLL on the rise: Swedish learner language corpus for European reference level studies. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 206–212, Portorož, Slovenia. European Language Resources Association (ELRA).

Elena Volodina, Ildikó Pilán, and David Alfter. 2016b. Classification of Swedish learner essays by CEFR levels. In *CALL communities and culture – short papers from EUROCALL 2016*, pages 456–461. Research-publishing.net.

Yu Wang, Yuelin Wang, Kai Dang, Jie Liu, and Zhuo Liu. 2021. A comprehensive survey of grammatical error correction. *ACM Trans. Intell. Syst. Technol.*, 12(5).

Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, Courtney Biles, Sasha Brown, Zac Kenton, Will Hawkins, Tom Stepleton, Abeba Birhane, Lisa Anne Hendricks, Laura Rimell, William Isaac, Julia Haas, Sean Legassick, Geoffrey Irving, and Iason Gabriel. 2022. Taxonomy of risks posed by language models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 214–229, New York, NY, USA. Association for Computing Machinery.

Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma, and Liang He. 2022. A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems*, 135:364–381.

Kevin P. Yancey, Geoffrey Laflair, Anthony Verardi, and Jill Burstein. 2023. Rating short L2 essays on the CEFR scale with GPT-4. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 576–584, Toronto, Canada. Association for Computational Linguistics.

Joey Öhman, Severine Verlinden, Ariel Ekgren, Amaru Cuba Gyllensten, Tim Isbister, Evangelia Gogoulou, Fredrik Carlsson, and Magnus Sahlgren. 2023. The nordic pile: A 1.2tb nordic dataset for language modeling. *arXiv preprint*, arXiv:2303.17183.

A Perplexity Plots for the Beginning of the Essays

In this appendix we present typical ’perplexity shapes’ for the beginning of a sentence. In Figure 4 we present the plots for the first 25 tokens of the essays from Figure 5 with the exception of the one at level C1.

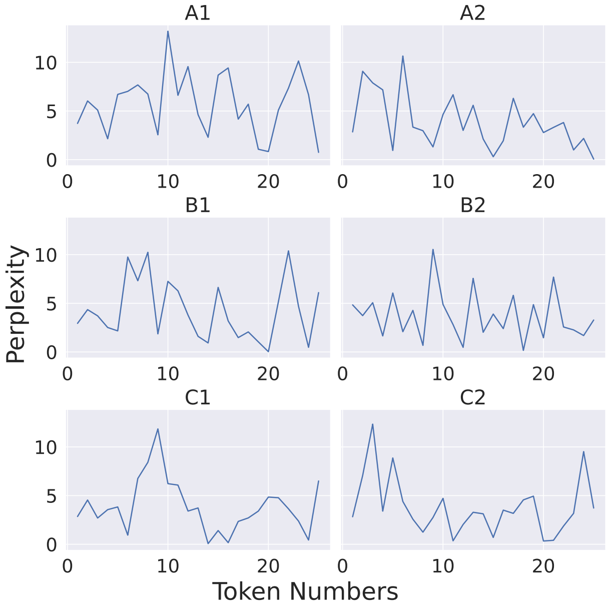


Figure 4: Perplexity plots for the first 25 tokens of the sample essays from Figure 5. The X-axis shows the running number of a token, while the Y-axis shows the perplexity score. Relative perplexity for the first several tokens is stably high, with a few exceptions. Essays at C1 and C2 level exhibit the same tendency.

B Perplexity Plots of the Essays

In Figure 5 we present plots of the perplexity changes throughout some of the essays. These plots were used to help inform a cut-off line between what we consider relatively high and relatively low perplexity values.

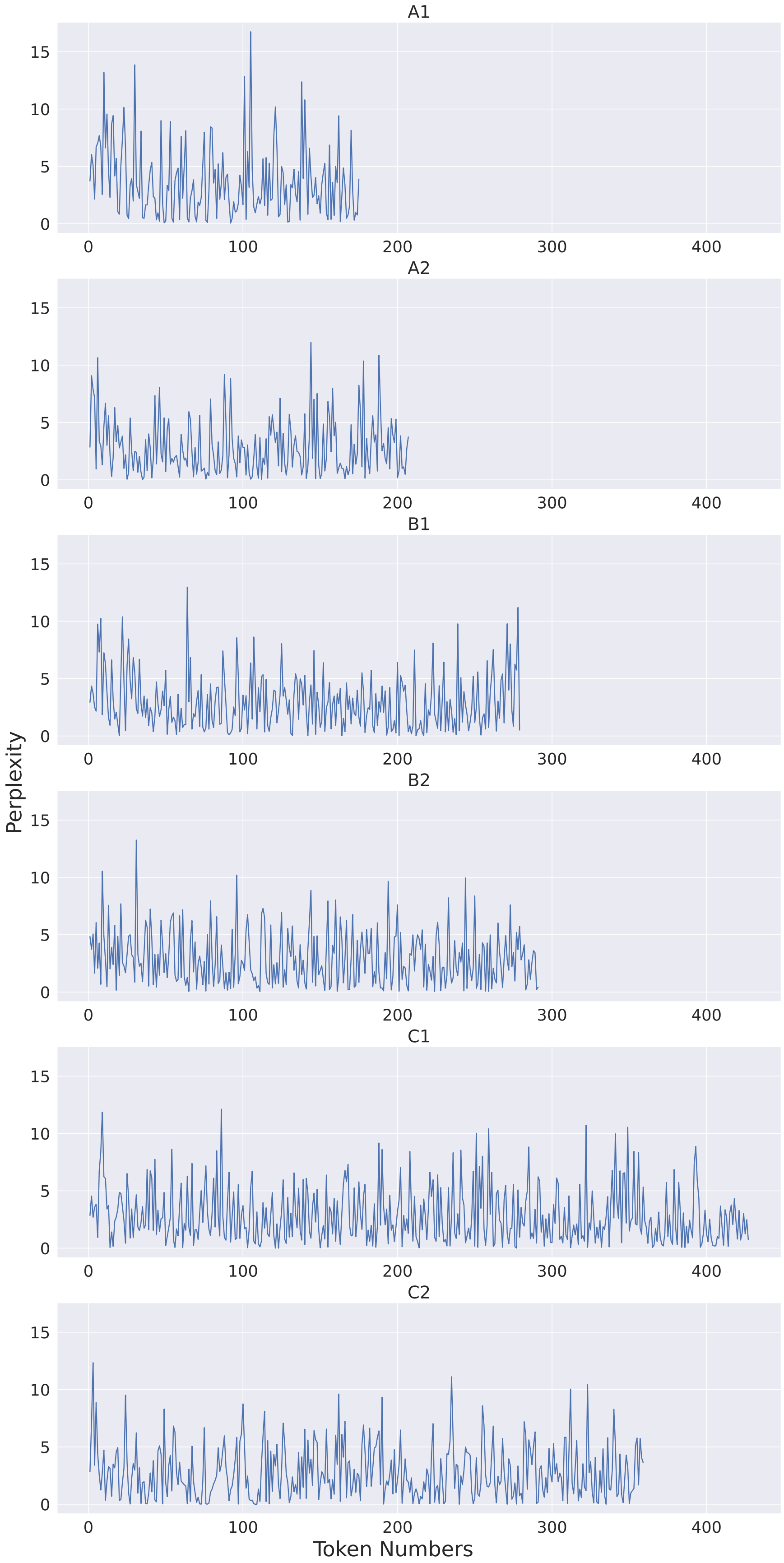


Figure 5: Sampled perplexity shapes for full essays with median perplexity at levels A1 to C1. The X-axis shows the running number of a token, while the Y-axis shows the perplexity score.

BERT-IRT: Accelerating Item Piloting with BERT Embeddings and Explainable IRT Models

Kevin P. Yancey and **Andrew Runge** and **Geoff LaFlair** and **Phoebe Mulcaire**
Duolingo
5900 Penn Ave
Pittsburgh, PA 15206
{kyancey,arunge,geoff,phoebe}@duolingo.com

Abstract

Estimating item parameters (e.g., the difficulty of a question) is an important part of modern high-stakes tests. Conventional methods require lengthy pilots to collect response data from a representative population of test-takers. The need for these pilots limit item bank size and how often those item banks can be refreshed, impacting test security, while increasing costs needed to support the test and taking up the test-taker's valuable time. Our paper presents a novel explanatory item response theory (IRT) model, BERT-IRT, that has been used on the Duolingo English Test (DET), a high-stakes test of English, to reduce the length of pilots by a factor of 10. Our evaluation shows how the model uses BERT embeddings and engineered NLP features to accelerate item piloting without sacrificing criterion validity or reliability.

1 Introduction

The Duolingo English Test (DET) is a test of English language proficiency that is used for admissions decisions in English medium universities. It measures the four skills of speaking, writing, reading, and listening. It is delivered remotely to test-takers' computers via a desktop application, and it can be taken any time and at any appropriate location with a strong enough internet connection. The DET's value proposition to test-takers is that it is affordable, short in duration, and has a short score reporting turn-around time.

The DET accomplishes this, in part, by using computer adaptive test (CAT) administration to more quickly and accurately estimate test-takers' language proficiency (Cardwell et al., 2022). A computer adaptive test (CAT) uses item parameter estimates to adapt to each test-taker by finding items that will yield maximal information about their proficiency based on how well they've done so far. Item banks for CATs must be very large to ensure that test-takers do not have preknowledge of items (LaFlair et al., 2022; Way, 1998), and they also require high-quality item parameter estimates to ensure that items are selected for administration accurately.

Typically, item parameters are estimated from hundreds of responses for each item collected via pilots. However, these pilots take up the test-taker's valuable time and increase the costs for the assessment, thus limiting the rate at which new items can be added to the bank. Explanatory frameworks that estimate item parameters from item features have been around for a long time, starting with Fischer (1973)'s Log Linear Traits Model (LLTM), and have a rich literature (De Boeck, 2004). These frameworks can be used to help reduce or eliminate the need for item piloting by leveraging item features to estimate item parameters more accurately with less response data. This can have positive downstream effects on test security and on test-takers. For security, it allows for test developers to add to, or replace, their item banks at very high rates, which helps to ensure unique administrations of tests and reduce the effects of item preknowledge. For test-takers, it reduces the amount of time they spend responding to unscored test items during pilots and reduces the costs of test development. These cost savings can be passed on to test-takers and even help lower barriers for less economically advantaged test-takers.

It is well known in the NLP literature (Tenney et al., 2019; Jawahar et al., 2019) that pretrained language models such as BERT (Devlin et al., 2019) learn text representations that represent highly general linguistic properties of words that are useful for a wide range of tasks, including estimating the difficulty of text for L2 learners (Yancey et al., 2021). More recent work has explored using these text embeddings in explanatory IRT models to predict parameters for test items. For example, Benedetto et al. (2021) finetuned

BERT to predict difficulty using datasets of educational questions and real student responses, and Byrd and Srivastava (2022) combined contextual embeddings from BERT with additional manually curated features to predict difficulty and discrimination for general knowledge questions. Similar work has used BERT to predict the difficulty of multiple-choice questions (Reyes et al., 2023) and programming problems (Zhou and Tao, 2020).

One example of using explanatory models this way is described in our previous work, McCarthy et al. (2021), which proposed using BERT embeddings in a multi-task explanatory item response theory (IRT) framework, called BERT-LLTM, to estimate the item parameters of c-test tasks, a task typically used to assess L2 language proficiency. This work introduces a new model, BERT-IRT, which makes several improvements to this approach:

- BERT-LLTM estimated passage-level difficulty and discrimination. BERT-IRT estimates these at the word level, which greatly improves criterion validity and reliability.

- The accuracy of BERT-LLTM's parameter estimates is limited by how well the features predict those parameters, even for items that have enough observed responses that non-explanatory IRT models could produce more accurate estimates. BERT-IRT achieves the best of both worlds by using residual weights, which allows it to refine the parameter estimates derived from features based on response data that has been collected for each item in a manner similar to Bayesian updating.

- BERT-IRT incorporates engineered NLP features that substantially increase the accuracy of the model's parameter estimates.

In addition to the offline evaluation on historical data, we present the results of using this model to shorten pilots by a factor of 10 on a real-world high-stakes test of English for L2 learners.

2 Background: Language Assessment

First, we will provide a brief overview of the relevant concepts from language assessment research.

2.1 Item Response Theory (IRT)

Item Response Theory (IRT; (Lord, 2012)) is essential for most modern high-stakes tests, and for Computer Adaptive Tests (CAT; (Weiss, 1982; Van der Linden and Glas, 2010)) in particular. IRT models are statistical models that are used to improve the time-efficiency and accuracy of assessment by modeling item characteristics (called "parameters") that affect the probability of test-takers of different proficiency levels responding to that item correctly. One of the most common IRT models is the 2PL model (Hambleton et al., 1991), which models both the relative difficulty of an item and how well an item discriminates between high and low proficiency test-takers. IRT models are used to quantify how informative an item will be for a given test-taker (i.e., by computing its Fisher information), which is used by CAT algorithms to increase the efficiency of the test. Additionally, IRT models are used to produce scores from CAT algorithms by computing the expected-a-posteriori (EAP) or maximum-a-posteriori (MAP) of the test-taker's latent proficiency based on the test-taker's observed responses to items and the estimated parameters for those items (Van der Linden and Glas, 2010).

2.2 Validity & Reliability

Validity and reliability are two key concepts in assessing the quality of scores (Furr, 2021), which are the main product of an assessment. Validity refers to the degree to which the score measures its intended "construct" (i.e., what it's intended to measure). One common piece of validity evidence is criterion validity, which is the test score's correlation with other known measures of the same or similar construct. Reliability is the consistency of the score. This is often measured by taking the correlation between retests by the same test-taker (i.e., test-retest reliability).

2.3 The C-Test Task Type

This paper focuses on estimating item parameters of c-test tasks for L2 learners of English. C-tests are reading tasks that measure test-takers' general language ability (Norris, 2018). As shown in Figure 1, each c-test task is composed of a paragraph in which some of the words are damaged by removing the second half of the word. Specifically, the first and last sentences of the passage are left intact to provide context, but every other word of the intermediary sentences is damaged. The test-takers' task is to complete all of the damaged words. Research on c-tests has shown that test-taker performance on these tasks correlates with overall language proficiency test scores (Daller et al., 2021), measures of reading ability (Khodadady, 2014; Klein-Braley, 1997), as well as vocabulary, and grammatical knowledge (Eckes and Grotjahn, 2006; Karimi, 2011; Khodadady, 2014).

2.4 Testlets

In our IRT model, we treat each damaged word as a distinct item with its own parameters. This essentially makes each c-test task a testlet (Wainer et al., 2007), where multiple items are administered together and share a common context (i.e., the passage). In our internal evaluation, we found that treating each damaged word as a distinct item dramatically increased criterion validity and reliability, as the IRT model was able to account for the differences in difficulty and discrimination among words within the passage. Specifically, using the Spearman-Brown prophecy formula (Allen and Yen, 2001), we found that we would have to add 25 % more c-test passages to each test session in order to achieve the same increase in test-retest reliability without using testlet scoring.

3 Model

In the following sub-sections, we explain the BERT-IRT model in detail, starting with explaining the standard 2PL IRT model in Section 3.1 and then extending it with an explanatory framework in Section 3.2. We then discuss the BERT-IRT model's features in Section 3.3, before finally explaining the training process in Section 3.4.

3.1 The Standard 2PL IRT Model

We start by formally defining the standard 2PL model, which is extended by our BERT-IRT model. In the 2PL model, the probability that a test-taker with proficiency θ_p will get item i correct depends on two item parameters:

- The intercept, denoted d_i, that models the logit-probability that a test-taker with average ability will answer the item correctly. This measures how easy or difficult the item is.
- The slope, denoted a_i, that defines how much that logit-probability changes depending on a test-taker's proficiency. This measures how discriminative the item is.

With these two item parameters, the 2PL model defines the probability of test-taker p getting item i correct as:

$$P(Y_{p,i} = 1) = f_{\text{logistic}}(d_i + a_i\theta_p)$$

where $Y_{p,i} \in \{0, 1\}$ is the test-taker's grade on the item.

3.2 Explanatory IRT Framework

In the standard 2PL model, each item parameter would be estimated by finding the values that best predict the observed responses for that item. As in other explanatory IRT frameworks, BERT-IRT extracts features from items and uses those features to predict item parameters as functions of those features. This has two key advantages:

1. This can reduce the amount of response data needed to estimate accurate parameters.
2. This allows one to estimate item parameters for novel items for which no response data has been collected.

However, for an item with many observed responses, explanatory IRT models may produce less accurate item parameter estimates than what could be achieved by non-explanatory IRT models, due to variance in item parameters that are not explained by the features. To overcome this, BERT-IRT uses residual weights to adjust the item parameter estimates of each item based on the observations for that particular item.

BERT-IRT uses a set of K item features to estimate a_i and d_i. Let $X_{i,k} \in \mathbb{R}$ denote the value of the k-th feature for item i where $X_{i,0}$ is a constant such that $X_{i,0} = 1$ for all i.

An item's intercept parameter, d_i, is thus modeled as a linear function of the item's features, X_i, plus the item-specific residual, denoted $\varepsilon_{d,i}$. The equation for d_i thus becomes:

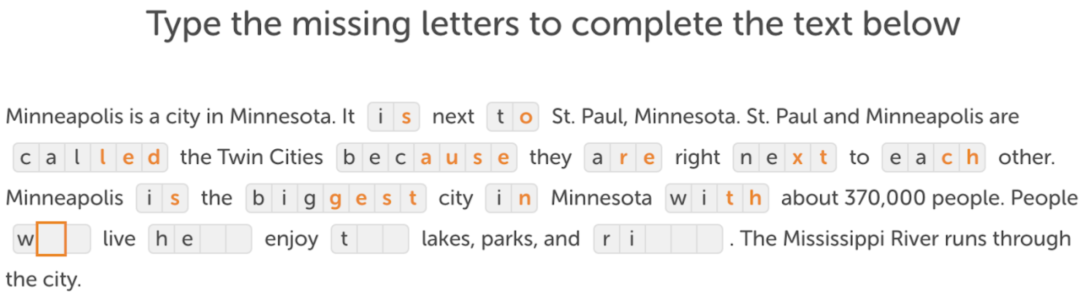
$$d_i = \varepsilon_{d,i} + \sum_{k=0}^{K} \upsilon_k X_{i,k}$$

where $\upsilon \in \mathbb{R}^{K+1}$ is a vector consisting of the bias term, υ_0, and the feature weights.

Slope parameters are defined similarly, but use a log-linear framework. The formula for slope parameters is thus:

$$a_i = \exp\left(\varepsilon_{a,i} + \sum_{k=0}^{K} \beta_k X_{i,k}\right)$$

where $\beta \in \mathbb{R}^{K+1}$ is the vector consisting of the bias term and feature weights, and $\varepsilon_{a,i}$ is the residual weight. The log-linear framework is often



Type the missing letters to complete the text below

Minneapolis is a city in Minnesota. It is next to St. Paul, Minnesota. St. Paul and Minneapolis are called the Twin Cities because they are right next to each other. Minneapolis is the biggest city in Minnesota with about 370,000 people. People w live he enjoy t lakes, parks, and ri . The Mississippi River runs through the city.

Figure 1: Example C-Test Item

closer to the true relationship between the slope parameters and the item features, has nicer convergence properties, and enforces that slope parameters are positive.

3.3 Model Features

Most of the features used by BERT-IRT are extracted from the pretrained BERT model by feeding in the undamaged passage (i.e., the passage without letters omitted from the damaged words). Two embeddings for each item are used as features:

Passage Embedding (n=768) - This is computed as the average of the embeddings extracted for each token in the passage from BERT's 11th layer.

Contextual Word Embedding (n=3,072) - This is computed by concatenating the token's embeddings from the first four layers of BERT. If the damaged word corresponds to multiple BERT tokens, then the embeddings for the applicable tokens are averaged.

Various alternative methods for encoding ctest items were evaluated in preliminary experiments, and this approach was found to be among the best. In particular, we found that using the lowest four layers of BERT to produce contextual word embeddings outperformed using higher layers. We believe this is because lower layers are better able to encode surface-level information, such as word frequency (Jawahar et al., 2019; Li et al., 2021), that are often important to predicting L2 difficulty (François and Fairon, 2012).

In addition, BERT-IRT uses 15 engineered NLP features shown to correlate strongly with c-test item parameters, specifically:

- The log frequency of the damaged word in the Corpus of Contemporary American English (COCA) (Davies, 2008)
- The log frequency of the word in COCA across the 8 sub-corpora (8 features)
- The log document frequency of the damaged word in the COCA corpus
- The length of the answer key (i.e., the number of letters the test-taker must fill in)
- The proportion of vowels in the answer key
- The average log frequency in COCA of each word in the c-test passage
- The position of the damaged word within the passage, normalized by the passage's length
- The conditional probability of the correct word, given the damaged word, derived using COCA frequencies (e.g. if the damaged word is "pass___" and the correct word is "passage", how frequently does that word occur versus alternative solutions such as "passing" vs. "passers" etc.)

3.4 Model Training

To estimate the model weights,[1] we need a training dataset of graded responses from test-takers. This consists of a set of test-taker responses represented as tuples of item, i, test-taker, p, and grade, $g \in \{0, 1\}$. We essentially use gradient descent to perform maximum-a-posteriori (MAP) estimation of the model weights given the observed response data. Details are provided in the subsections below.

3.4.1 Model Weights

The model has four vectors of weights that must be estimated: the intercept bias and feature weights vector, $\upsilon \in \mathbb{R}^{K+1}$, the intercept residuals vector,

[1] Here, we refer to all of the model's learnable parameters as weights to avoid them being conflated item parameters.

$\varepsilon_d \in \mathbb{R}^I$, the slope bias and feature weights vector, $\beta \in \mathbb{R}^{K+1}$, and slope residuals vector, $\varepsilon_a \in \mathbb{R}^I$, where I denotes the number of items in the training dataset.

3.4.2 Theta Estimates

Since our response data is collected as part of a high-stakes test of English, we can compute accurate estimates for test-taker proficiency based on their performance on items whose parameters are not being estimated (i.e., the section scores for item types other than c-test). We use these as fixed estimates for θ_p during model training. In other piloting designs where this is not possible, we could treat these proficiencies as weights to be estimated jointly with the other model weights, but that would require larger quantities of response data to achieve comparable performance results.

3.4.3 Regularization

To avoid the model being underidentified, the residual weights must be regularized. We apply L2 regularization to these parameters. Optimizing the strength of those L2 penalties is important: if the L2 penalties are set too low then the model won't generalize to new items as well as it could, and if they are set too high the model will predict item parameters for items with many observations less accurately than it could. In this context, these L2 penalties are equivalent to using Gaussian priors with zero means. The optimal penalty for intercept residuals would be $0.5/\sigma^2_{\varepsilon_d}$, where $\sigma^2_{\varepsilon_d}$ is the variance in the intercepts that is *not* explained by the features. The optimal penalty for slope residuals is likewise. Thus, we treat $\hat{\sigma}^2_{\varepsilon_d}$ and $\hat{\sigma}^2_{\varepsilon_a}$ as hyperparameters, and set the penalties for intercept residuals and slope residuals to $0.5/\hat{\sigma}^2_{\varepsilon_d}$ and $0.5/\hat{\sigma}^2_{\varepsilon_a}$, respectively.

Since there are many features, we also use L2 regularization on the feature weights. Following the same convention, we set the coefficients of these penalties as $0.5/\hat{\sigma}^2_{\beta}$ and $0.5/\hat{\sigma}^2_{\upsilon}$, respectively, treating $\hat{\sigma}^2_{\beta}$ and $\hat{\sigma}^2_{\upsilon}$ as hyperparameters.

3.4.4 Training Objective

During training, we initialize all weights to zero and use gradient descent to estimate values for the model weights that maximize their log posterior-probability given the test-taker responses in the training dataset, D. The objective function to be maximized is thus specified as follows:

$$\sum_{(i,p,g)\in D} LL(\Phi \mid Y_{p,i} = g) - \frac{0.5}{\hat{\sigma}^2_{\upsilon}} \sum_{k=1}^{K} \upsilon_k^2 \\ - \frac{0.5}{\hat{\sigma}^2_{\beta}} \sum_{k=1}^{K} \beta_k^2 - \frac{0.5}{\hat{\sigma}^2_{\varepsilon_a}} \sum_{i=1}^{I} \varepsilon^2_{a,i} - \frac{0.5}{\hat{\sigma}^2_{\varepsilon_d}} \sum_{i=1}^{I} \varepsilon^2_{d,i}$$

where Φ denotes the set of weight vectors being estimated (β, υ, ε_a, and ε_d) and LL is the log likelihood function:

$$LL(\Phi \mid Y_{p,s}) = g \cdot \ln P(Y_{p,i} = 1) \\ + (g-1) \cdot \ln(1 - P(Y_{p,s} = 1))$$

3.4.5 Tuning Hyperparameters

The large search space resulting from four hyperparameters and long training times makes tuning hyperparameters difficult. For our experiments, we used a sparse grid search to find acceptable values for hyperparmeters. Since the optimization of the residual hyperparameters requires evaluating how well the model predicts both novel and seen items, we ensured that the training and evaluation datasets were split in such a way that the evaluation dataset included both items that occurred in the training dataset and items that did not.

We found that even a limited search of the hyperparameter space produced good results. However, there are methods that could be applied to the training data to estimate $\sigma^2_{\varepsilon_a}$ and $\sigma^2_{\varepsilon_d}$ directly. These include maximizing the marginal likelihood function, maximizing an approximation to the marginal likelihood function, and fully Bayesian methods implemented via Markov Chain Monte Carlo (Dey et al., 1997; Lindstrom and Bates, 1990; Pinheiro and Bates, 1995; Wolfinger, 1993). Future work could consider the application of these methods.

4 Experiments

Here we present a series of four experiments to evaluate BERT-IRT using data from the Duolingo English Test, a high-stakes test of English for L2 learners. In the first experiment, we use offline evaluation to analyze the model's performance when piloting a new item bank from scratch (i.e., what we refer to as a "fast-start" scenario). In the second experiment, we analyze BERT-IRT's ability to generalize item parameter predictions to unseen items under various conditions. In the third experiment, we investigate how much each feature contributes to the estimation of item parameters. Finally, in the forth experiment, we analyze BERT-IRT's ability to leverage response data from an existing item bank to make predictions for new items with limited piloting data available (i.e., what we refer to as a "jump-start" scenario). As part of this, we discuss the results of using BERT-IRT to add new items to the test's item bank with only a tenth of the normal amount of pilot data.

4.1 Experiment 1. Offline Evaluation in a Fast-Start Scenario

In this experiment, we do an ablation study to evaluate how the model performs when only a limited amount of response data is available for each item. Traditionally, a new item bank would be piloted until 200 observations per item are collected (the minimum needed for reasonably accurate item parameters in an unregularized 2PL model). However, these pilots can be costly and time-consuming, so with BERT-IRT we hope to be able to achieve similar or better performance with much shorter pilots.

For this experiment, we retrieved around a year's worth of historical response data from the test. The dataset included around 3,000 c-test passages with around 50,000 unique items. The unablated dataset had around 600 observations per item, which were split into train and evaluation datasets. The training dataset was sampled to produce ablated training datasets with observation counts of 5, 10, 20, 40, 80, 160, and 200 observations per item.

We compared BERT-IRT to two baselines:

Post-Pilot Operational 2PL - A non-explanatory 2PL model trained on 200 responses per item (i.e., the minimal number of responses per item collected during a standard pilot). This simulates the performance of using the test's operational 2PL model on items that have only recently been created and piloted.

Regularized 2PL - A non-explanatory 2PL model trained on the same ablated datasets as BERT-IRT, where the item parameters are estimated via maximum-a-posteriori (MAP) estimation using a Gaussian prior on each parameter. This regularization is used because unregularized 2PL models will yield very poor results when trained on fewer than 200 responses per item.

We then used those trained models to produce probabilities and scores on the evaluation dataset, which we evaluated using the following metrics:

Cross-Entropy - The cross-entropy between observed binary grades and their probability as predicted by the IRT model. This measures how well the model predicts the probability of the test-taker responding to an item correctly.

Item Mean Grade R - The Pearson correlation between each item's observed mean grade in the response dataset and it's predicted mean-grade according to the IRT model. This mainly measures the IRT model's ability to predict the relative difficulty of each item.

Test-Retest Correlation - The Pearson correlation between c-test scores produced by the IRT model for any two test sessions taken by the same test-taker within 30 days of eachother. This is a well established measure of score reliability in the assessment research literature (Furr, 2021).

Internal Validity Coefficient - The Pearson correlation between the c-test score produced by the IRT model, and the score aggregated from other sections of the test (using their original scoring methods). This is a common measure that is used in the assessment research literature (Furr, 2021) to measure criterion validity.

The results are shown in Figure 2. These plots show that the BERT-IRT model always outperformed the regularized 2PL model regardless of the number of responses available for training. Furthermore, these results show that the BERT-IRT model can achieve similar or better performance than the operational 2PL model with as few as 50 responses per item, representing a *4X increase* in piloting efficiency. The only metric on which BERT-IRT did not outperform the Post-Pilot Operational 2PL baseline was the Internal Validity Coefficient. However, given that this is the case even when BERT-IRT's test-retest reliability is higher, this could indicate the BERT-IRT is finding parameters that better represent aspects of the construct that are specific to c-test items. This could increase test-retest reliability by more accurately measuring the skills needed to answer c-test items, but lower internal validity because the skills measured by c-tests are slightly different than those measured

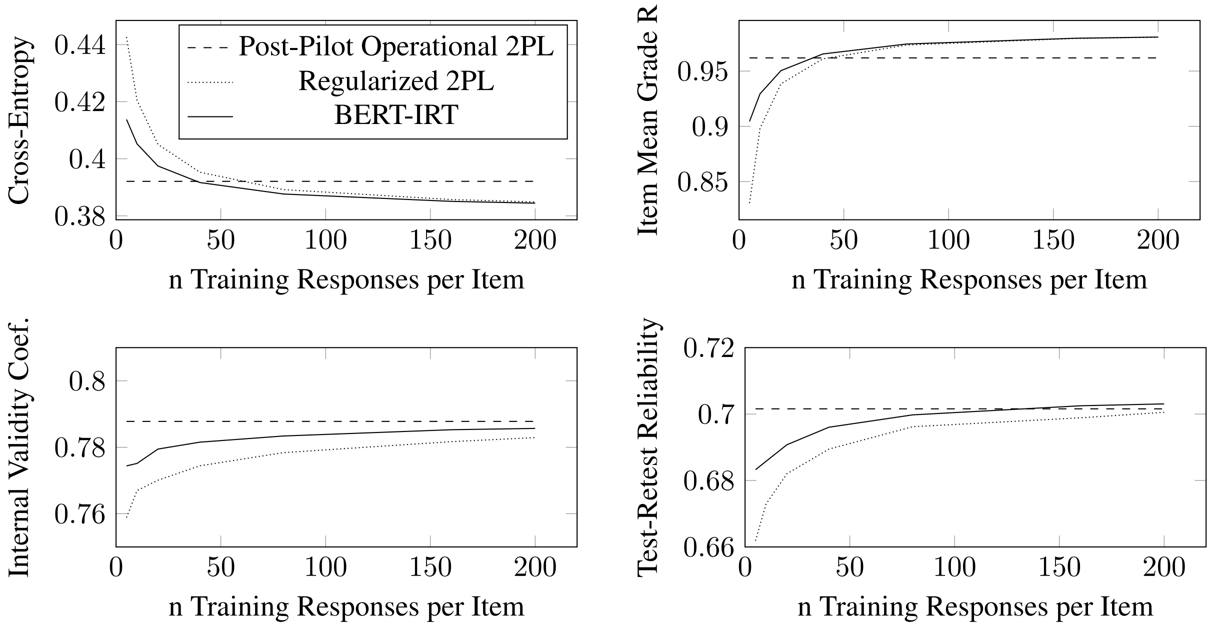


Figure 2: Experiment 1. Evaluation in a Fast-Start Scenario

by other task types. In any case, the difference in internal validity is very small (0.78 vs 0.79).

Split	Cross-Entropy	Item Mean Grade R
Test-Taker	0.38	0.98
Testlet	0.43	0.88
Item	0.42	0.89
Stem	0.52	0.76

Table 1: Comparison of BERT-IRT item parameter estimates when trained on 20 vs 200 responses.

4.2 Experiment 2. Generalization Experiments

To better understand how the model generalizes parameter estimates across items, we experiment with different splits of the same dataset used in Experiment 1. The splits we used are defined below:

Test-Taker - Response data is split such that all responses for an individual test-taker are assigned to either the training or evaluation datasets. This simulates a fast-start scenario, as in Experiment 1. Since all items occur in training, this is essentially a baseline indicating the ceiling of what should be possible.

Testlet - Response data is split such that all responses to a given c-test passage (i.e., testlet) are assigned to either the training or evaluation datasets. This simulates a jump-start scenario, whereby responses for an existing item bank are used to estimate parameters for new items that have little or no pilot data.

Item - Response data is split such that all responses to a given item are assigned to either training or evaluation datasets. This investigates how well the model can predict item parameters for words in a passage, when there is significant response data for other words in the same passage. This might be useful if one wanted to change which words in a passage are damaged based on its predicted item parameters in order to adjust the c-test passage's difficulty or increase is informativeness.

Stem - Responses data is split such that all responses for items that share a word stem are assigned to either training or evaluation datasets. For example, items for "work", "worked", and "works" would all be put on the same side of the split. For this purpose, we used the Snowball Stemmer from NLTK (Porter, 1980; Bird et al., 2009). This evaluates how well the model generalizes to items assessing previously untested words.

In all cases, we use roughly 80 % of the data for training and 20 % for evaluation. Since under these data splits, individual sessions are split across training and evaluation datasets, its not possible to compute scores for sessions using just evaluation data. Hence, for this experiment we use only metrics that can be computed for individual item responses: Cross-Entropy and Item Mean Grade R.

The results are shown in Table 1. In the baseline split, the model almost perfectly predicts the mean grade of each item over the evaluation dataset, with a correlation of 0.98. The testlet and item splits shows that BERT-IRT generalizes very well to unseen items, predicting the mean grades of unseen items with a correlation of 0.88.

Notably, as shown by the stem split, the model's ability to predict mean grades for a item degrades significantly when that item has a novel stem that the model did not see in training. This shows that the item's word stem explains a significant amount of the variance in the item's parameters. This is a very useful property when jump-starting item parameters using BERT-IRT, because, due to Zipf's law, if the existing item bank is sizeable, most items of newly-created c-test passages will likely share a word-stem with an existing item from the existing bank. However, this means items with novel word stems will likely have less accurate item parameter estimates until sufficient response data for them can be collected.

4.3 Experiment 3. Feature Contributions

To better understand the contributions of various features, we evaluated the importance of each feature using SHAP values (Lundberg and Lee, 2017). In the BERT-IRT model, the features only affect the item parameter estimates through a linear combination defined by the weight vectors υ and β. As such, we compute the SHAP values using the same methods as would be used for linear models using those weight vectors. To account for correlations among features, we compute *observational* SHAP values. From these we compute the feature importance for each feature as the mean absolute SHAP value over all items, and then normalize the resulting feature importances to sum to 1. Since embeddings consist of hundreds of features that would be impractical to list individually, we summarize their importances by summing the embedding feature SHAP values for a given item before taking the absolute value and averaging across items. We also summarize the 8 genre-specific word-frequency features the same way.

The results are shown in Figure 3. The features are presented in the same order as in Section 3.3. For predicting both intercept parameters and log slope parameters, the word embedding is very important, contributing 28 % and 40 % of the prediction, respectively. By comparison, passage embeddings are a relatively weak predictor, contributing only 3 % and 8 % of the prediction, respectively. The word frequency features are also a very important predictor, contributing even more than the word embedding does for predicting intercepts.

Features	Cross-Entropy	Item Mean Grade R
All Features	0.43	0.88
Embeddings	0.44	0.84
Engineered	0.48	0.69

Table 2: Comparison of BERT-IRT performance on the Testlet split when using different feature sets.

Additionally, we did an ablation study by repeating the Testlet split experiment from Experiment 2, but using only embedding features or only engineered features (see Table 2). These results show that while the embedding features perform quite well on their own, both sets of features complement each other to yield superior results.

4.4 Experiment 4. Online & Offline Evaluation in a Jump-Start Scenario

In this experiment, we evaluate how well BERT-IRT can estimate item parameters for a new pool of c-test items with only a very short pilot, when leveraging large amounts of response data from an existing item bank to learn the relationships between the item features and item parameters.

To test this scenario, we generated 1,039 new c-test passages with GPT-3 (Brown et al., 2020), and piloted them on the test, with each test session being randomly assigned one unscored pilot c-test task in addition to its normal 4 scored c-test tasks. We ran the pilot until we had collected around 20 responses per item. We trained BERT-IRT on both the response data from the existing bank and the pilot, and estimated the parameters for all the new items. In an offline evaluation, we showed that even if we'd used the existing BERT-IRT parameter estimates to score the pilot c-test tasks instead of one of the other 4 operational c-test tasks, criterion validity and reliability would have been negligibly affected. Based on that offline evaluation, we added the new c-test tasks to the operational bank, replacing roughly a third of the existing c-test item bank with only a tenth of the piloting time that would have otherwise been required. Furthermore, analyses of the test following the item bank change

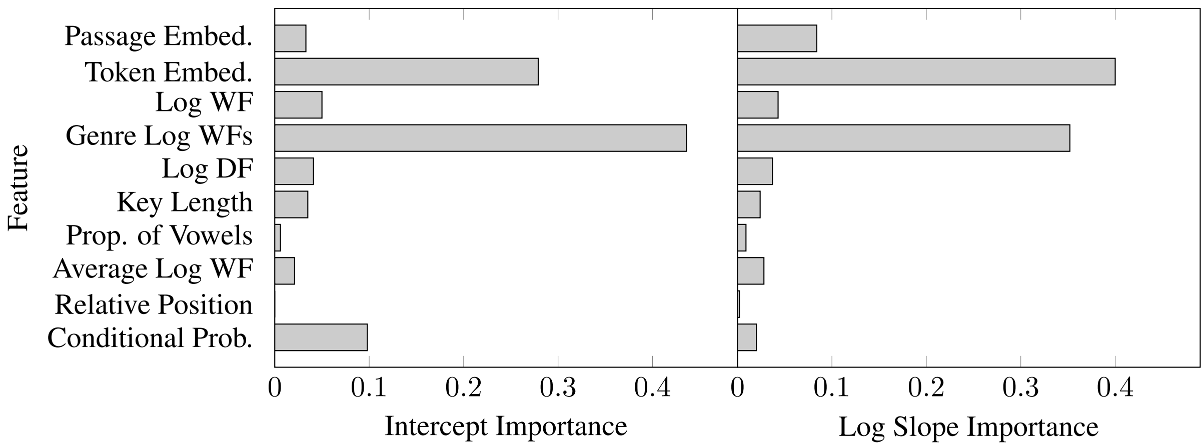


Figure 3: Feature Importances

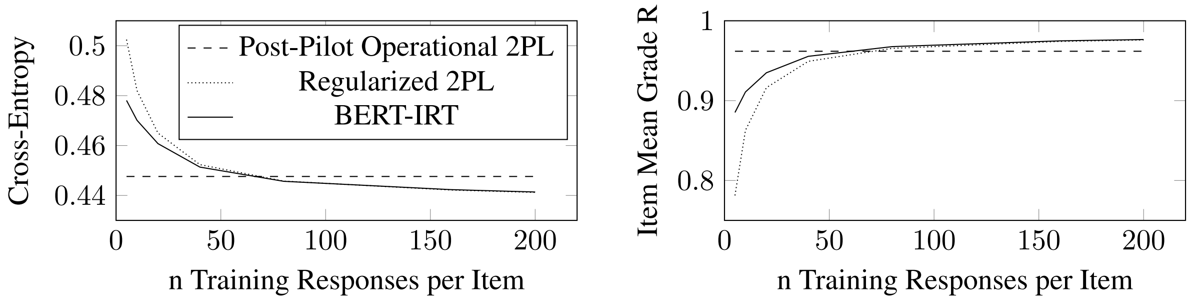


Figure 4: Experiment 4. Evaluation in a Jump-Start Scenario

confirmed that there was no significant impact on criterion validity or reliability.

Since adding the items to the operational item bank, we have collected substantial response data for all the new items, and are able to evaluate the quality of item parameters that would have been obtained had they been estimated with more data. To that end we conducted an ablation experiment similar to Experiment 1, but in a jump-start scenario (i.e., only the response data for the newly added items was ablated).

Figure 4 shows the results for Cross-Entropy and Item Mean Grade R for this ablation study. Similar to the results in Experiment 1, BERT-IRT outperformed the operational 2PL model with only a third of the data. As expected, it also out-performed the regularized 2PL model when trained on the same responses data. Importantly, even though a full third of the c-test item bank was replaced, this ablation study indicates that the impact on criterion validity and reliability would be negligible even if as few as 5 responses per item had been collected (i.e., the maximum difference between BERT-IRT and the Post-Pilot Operational IRT was less than 0.001 for both the Internal Validity Coefficient and Test-Retest Reliability metrics, even when BERT-IRT was trained on as few as 5 responses for each of the new items). This finding stands to dramatically boost the rate at which the item bank can be refreshed.

5 Conclusion & Future Work

In this paper, we demonstrated how an explanatory IRT model with BERT embeddings and other engineered NLP features can be used to accurately estimate item parameters for c-test items with limited piloting data. We showed that the model is able to use these features to generalize item parameter estimates across items, and that both BERT embeddings and engineered features contribute to the performance of the model. Furthermore, we showed how this was used on a high-stakes test of English to replace a third of its item pool with a tenth of the data that would normally have been required. Finally, our ablation study in Experiment 4 showed that we should be able to use BERT-IRT to reduce the pilot even further with negligible impact on criterion validity or reliability.

In a future work, we plan to explore similar applications of NLP and explanatory IRT models to other item types, and ways to reduce or eliminate the need for item piloting even further.

6 Limitations

There are three main limitations to our study:

- As mentioned in Section 3.4.5, this method could be improved if one were to incorporate a method to directly estimate the variance in item parameters that is explained by the features. However, finding a method that is tractable for a large number of features is difficult, and so we leave that to a future work.

- This study only evaluated the model on c-test tasks. Applications to other task types will need to be evaluated, and may require different features or IRT models to achieve good results.

- While Experiment 4 showed we successfully added a large number of c-test items to the bank with as few as 20 pilot responses per item, the ablation study that indicates we may be able to use even fewer pilot responses does not account for the potential impact that less accurate item parameters could have on the efficiency of the CAT algorithm. While we expect that impact would not significantly change our results, more study is needed to ensure that items could safely be added to the test with fewer than 20 responses per item.

Acknowledgements

We would like to thank all of those who helped review this paper and those who provided input on the development of BERT-IRT over the years, especially Alina von Davier, J.R. Lockwood, Burr Settles, and Jill Burstein.

References

Mary J Allen and Wendy M Yen. 2001. *Introduction to measurement theory*. Waveland Press.

Luca Benedetto, Giovanni Aradelli, Paolo Cremonesi, Andrea Cappelli, Andrea Giussani, and Roberto Turrin. 2021. On the application of transformers for estimating the difficulty of multiple-choice questions from text. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 147–157, Online. Association for Computational Linguistics.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

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*.

Matthew Byrd and Shashank Srivastava. 2022. Predicting difficulty and discrimination of natural language questions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 119–130, Dublin, Ireland. Association for Computational Linguistics.

Ramsey Cardwell, Ben Naismith, Geoffrey T LaFlair, and Steven Nydick. 2022. Duolingo English Test: Technical Manual. Duolingo Research Report, Duolingo.

Michael Daller, Amanda Müller, and Yixin Wang-Taylor. 2021. The C-test as predictor of the academic success of international students. *International Journal of Bilingual Education and Bilingualism*, 24(10):1502–1511.

Mark Davies. 2008. Word frequency data from The Corpus of Contemporary American English (COCA). `https://www.wordfrequency.info`.

Paul De Boeck. 2004. *Explanatory item response models: A generalized linear and nonlinear approach*. Springer Science & Business Media.

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.

Dipak K Dey, Ming-Hui Chen, and Hong Chang. 1997. Bayesian approach for nonlinear random effects models. *Biometrics*, pages 1239–1252.

Thomas Eckes and Rüdiger Grotjahn. 2006. A closer look at the construct validity of C-tests. *Language Testing*, 23(3):290–325.

Gerhard H Fischer. 1973. The linear logistic test model as an instrument in educational research. *Acta psychologica*, 37(6):359–374.

Thomas François and Cédrick Fairon. 2012. An "ai readability" formula for french as a foreign language.

In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 466–477.

R Michael Furr. 2021. *Psychometrics: an introduction*. SAGE publications.

Ronald K Hambleton, Hariharan Swaminathan, and H Jane Rogers. 1991. *Fundamentals of item response theory*, volume 2. Sage.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3651–3657. Association for Computational Linguistics.

Neda Karimi. 2011. C-test and vocabulary knowledge. *Language Testing in Asia*, 1(4):7.

E. Khodadady. 2014. Construct validity of C-tests: A factorial approach. *Journal of Language Teaching and Research*, 5.

C. Klein-Braley. 1997. C-Tests in the context of reduced redundancy testing: an appraisal. *Language Testing*, 14(1):47–84.

Geoffrey T. LaFlair, Thomas Langenfeld, Basim Baig, André Kenji Horie, Yigal Attali, and Alina A. Davier. 2022. Digital-First Assessments: A Security Framework. *Journal of Computer Assisted Learning*, page jcal.12665.

Bai Li, Zining Zhu, Guillaume Thomas, Yang Xu, and Frank Rudzicz. 2021. How is BERT surprised? layerwise detection of linguistic anomalies. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4215–4228. Association for Computational Linguistics.

Mary J Lindstrom and Douglas M Bates. 1990. Nonlinear mixed effects models for repeated measures data. *Biometrics*, pages 673–687.

Frederic M Lord. 2012. *Applications of item response theory to practical testing problems*. Routledge.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Arya D. McCarthy, Kevin P. Yancey, Geoffrey T. LaFlair, Jesse Egbert, Manqian Liao, and Burr Settles. 2021. Jump-starting item parameters for adaptive language tests. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 883–899, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

John Norris. 2018. *Developing C-tests for estimating proficiency in foreign language research*. Peter Lang, Berlin, Germany.

José C Pinheiro and Douglas M Bates. 1995. Approximations to the log-likelihood function in the nonlinear mixed-effects model. *Journal of computational and Graphical Statistics*, 4(1):12–35.

Martin F Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

Diego Reyes, Abelino Jimenez, Pablo Dartnell, Séverin Lions, and Sebastián Ríos. 2023. Multiple-choice questions difficulty prediction with neural networks. In *International Conference in Methodologies and intelligent Systems for Techhnology Enhanced Learning*, pages 11–22. Springer.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Wim J Van der Linden and Cees AW Glas. 2010. *Elements of adaptive testing*, volume 10. Springer.

Howard Wainer, Eric T Bradlow, and Xiaohui Wang. 2007. *Testlet response theory and its applications*. Cambridge University Press.

Walter D Way. 1998. Protecting the integrity of computerized testing item pools. *Educational Measurement: Issues and Practice*, 17(4):17–27.

David J Weiss. 1982. Improving measurement quality and efficiency with adaptive testing. *Applied psychological measurement*, 6(4):473–492.

Russ Wolfinger. 1993. Laplace's approximation for nonlinear mixed models. *Biometrika*, 80(4):791–795.

Kevin Yancey, Alice Pintard, and Thomas Francois. 2021. Investigating readability of french as a foreign language with deep learning and cognitive and pedagogical features. *Lingue e linguaggio*, 20(2):229–258.

Ya Zhou and Can Tao. 2020. Multi-task bert for problem difficulty prediction. In *2020 International Conference on Communications, Information System and Computer Engineering (CISCE)*, pages 213–216.

Transfer Learning of Argument Mining in Student Essays

Yuning Ding[1], **Julian Lohmann**[2], **Nils-Jonathan Schaller**[3],
Thorben Jansen[3], **Andrea Horbach**[1,4]
[1]CATALPA, FernUniversität in Hagen, Germany
[2]Institute for Psychology of Learning and Instruction, Kiel University, Germany
[3]Leibniz Institute for Science and Mathematics Education at the University of Kiel, Germany
[4]Hildesheim University, Germany

Abstract

This paper explores the transferability of a cross-prompt argument mining model trained on argumentative essays authored by native English speakers (EN-L1) across educational contexts and languages. Specifically, the adaptability of a multilingual transformer model is assessed through its application to comparable argumentative essays authored by English-as-a-foreign-language learners (EN-L2) for context transfer, and a dataset composed of essays written by native German learners (DE) for both language and task transfer. To separate language effects from educational context effects, we also perform experiments on a machine-translated version of the German dataset (DE-MT). Our findings demonstrate that, even under zero-shot conditions, a model trained on native English speakers exhibits satisfactory performance on the EN-L2/DE datasets. Machine translation does not substantially enhance this performance, suggesting that distinct writing styles across educational contexts impact performance more than language differences.

1 Introduction

Argumentative writing is a central skill to succeed across school subjects (Graham et al., 2020) and automated feedback is an effective way to foster writing skills (Fleckenstein et al., 2023). Figure 1 shows an example of providing students with feedback by highlighting different argumentative elements, such as *lead, position, claim* and *conclusion* in their writing. Such feedback offers guidance to students for enhancing the structure of their essays.

However, training a dedicated feedback model for each new task could incur substantial costs. One approach to mitigate this expense is to transfer a pre-trained model to new datasets. While existing research highlights model transferability across different writing prompts (Ding et al., 2022), no research demonstrates whether employing English argument mining models across languages and different educational contexts yields consistent performance.

Such educational contexts for argumentative writing can be specified according to two dimensions: native vs. foreign language instruction on the one hand and independent vs. integrated writing tasks on the other hand.

We first have a closer look at the differences between L1 and L2 writing. In L1 teaching contexts, the emphasis is primarily on content, whereas in L2, the focus is on language acquisition and structure. These differences are also reflected in distinct cognitive models and therefore writing styles (Devine et al., 1993). Beyond obvious characteristics such as spelling and grammar errors in L2 writing, like the misspelled *advetisments* and the subject-verb disagreement exemplified in Figure 1, prior studies also unveiled that non-native English writers tend to craft shorter sentences and employ fewer hedges (e.g., *probably*, *may*) to moderate the strength of their claims, in contrast to native English speakers (Burrough-Boenisch, 2002). Moreover, L2 writers prefer a more straightforward argumentation structure and often avoid counter-arguments (Sanders and Schilperoord, 2006).

As for the second dimension, in independent tasks, individuals are typically provided with a specific writing prompt and are required to formulate their essays based solely on their thoughts, experiences, and knowledge. For example, the independent writing prompt of EN-L2 in Figure 1 is:

> *Do you agree or disagree with the following statement? Television advertising directed toward young children (aged two to five) should not be allowed. Use specific reasons and examples to support your answer.*

In integrated tasks, writers are presented with one or more texts related to a particular topic and then asked to synthesize information from the provided texts and incorporate it into their writing. For in-

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 439–449
June 20, 2024 ©2024 Association for Computational Linguistics

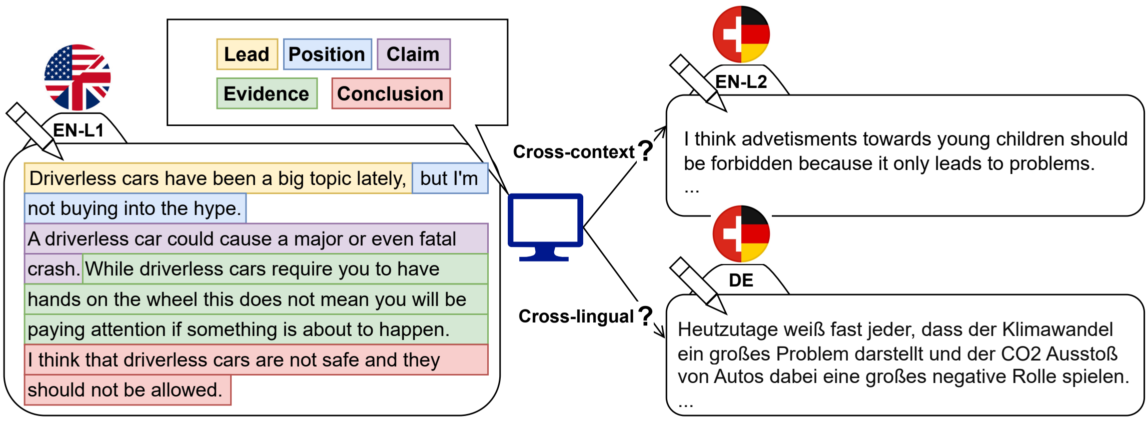

Lead
Position
Claim
Evidence
Conclusion
EN-L1
Driverless cars have been a big topic lately, but I'm not buying into the hype. A driverless car could cause a major or even fatal crash. While driverless cars require you to have hands on the wheel this does not mean you will be paying attention if something is about to happen. I think that driverless cars are not safe and they should not be allowed.
Cross-context?
Cross-lingual?
EN-L2
I think advetisments towards young children should be forbidden because it only leads to problems.
...
DE
Heutzutage weiß fast jeder, dass der Klimawandel ein großes Problem darstellt und der CO2 Ausstoß von Autos dabei eine großes negative Rolle spielen.
...

Figure 1: Example of automatic feedback provided by an argument mining model trained on EN-L1 essays and its uncertain transferability to EN-L2 (upper right) and DE (lower right) data.

stance, the DE dataset has an integrated writing prompt, which discusses using renewable energy sources to combat climate change. It presents three options: a wind farm, a solar park, and a hydropower plant. Students are asked to evaluate these options based on specific criteria, taking a stand in favor of one source and providing supporting arguments. Earlier studies have shown that task type can influence lexical complexity and argument structure in essays (Cumming et al., 2005; Guo et al., 2013).

Targeting the challenge of transfer learning brought by the differences described above and the languages, the following research questions are investigated in this paper:

RQ. 1 *How do linguistic structures and argumentation styles differ among English L1, L2, and German datasets?*

RQ. 2 *How can the argument mining model, initially trained on the English L1 dataset, be effectively transferred to L2 and German datasets? How much data is needed to achieve the best transfer performance?*

RQ. 3 *In the context of cross-lingual transfer, what roles are played by language differences and task disparities in influencing the model's performance?*

Through a comparative analysis of English L1, L2, and German datasets, we answer **RQ 1** by showing the statistical and structural distinctions inherent in argumentative essays across different educational contexts and languages. While the English L1 and L2 data are written for independent tasks, the German dataset is collected from integrated writing tasks, this completes our study with a focus not only on cross-lingual transfer learning but also on cross-educational contexts. We then conduct two experimental studies to transfer argument mining models trained on a large English L1 dataset to the L2 and German datasets for **RQ 2**. In addressing the challenge of cross-language transfer in **RQ 3**, our research extends to experiments involving the machine-translated version of the German dataset. This expanded scope enables a more profound examination of the variances in model performance arising from linguistic disparities and diverse writing tasks.

The answer to these questions could be invaluable in developing educational applications: with the appropriate adjustments, models trained on English L1 data can effectively be transferred to an L2 dataset. This would greatly benefit the development of educational applications, particularly in contexts where resources are limited, by providing students with access to high-quality learning tools and feedback systems. Additionally, the impact of linguistic differences on the model's effectiveness is essential for the development of educational applications aimed at student populations from different linguistic backgrounds, ensuring they receive the support they need to improve their argumentative writing skills.

2 Related Work

Transfer learning has been extensively studied for many years. Surveys such as Pan and Yang (2009), Weiss et al. (2016), and Zhuang et al. (2020) provide a comprehensive overview of the developments in this area over the years. Similarly, numerous studies have explored the topic of argument mining through literature reviews, evidenced by works like Peldszus and Stede (2013) and Lawrence and Reed (2020). In this paper, we focus our review of related work specifically on transfer learning within the educational domain and argument mining in student essays.

2.1 Transfer Learning in Education

In many educational scoring tasks, transfer learning is important in avoiding retraining a model for every new task. Especially in the area of automated essay scoring, cross-prompt and prompt-independent models are widely researched (e.g. Jin et al. (2018), Ridley et al. (2021), Xue et al. (2021))

Fewer approaches have focused on a transfer between languages in educational scoring, for example for content scoring (Horbach et al., 2018, 2023) or language proficiency classification (Vajjala and Rama, 2018).

Approaches for cross-lingual argument mining in the educational domain are even scarcer. Eger et al. (2018) automatically translated an educational argument mining dataset into various languages showing the feasibility of a cross-lingual transfer. To the best of our knowledge, we are the first to attempt such a transfer on authentic ecologically valid cross-lingual data, extending the research body on cross-lingual argument mining approaches in other domains such as medicine (Yeginbergenova and Agerri, 2023) or general controversial topics (Toledo-Ronen et al., 2020).

Differences in the educational and cultural context of argumentative essay scoring have been studied by Chen et al. (2022) finding that, for the ICLE corpus containing essays by English learners with 16 native languages, culture influenced learners' argumentation patterns substantially.

2.2 Argument Mining in Student Essays

Various approaches for argument mining in student essays exist with many of them adopting the persuasive essay scheme introduced by Stab and Gurevych (2014), such as Wambsganss et al. (2020); Putra et al. (2021) and Alhindi and Ghosh (2021). This model comprises four key categories: *major claim, claim, premise,* and *non-argumentative elements*.

In this study, we have five different argumentative elements, namely *lead, position, claim, evidence,* and *conclusion*, which is a simplified version of the task definition set by the Kaggle Feedback Prize competition [1] on the PERSUADE dataset (Crossley et al., 2022). This dataset adopts a variant of the Toulmin argument mining model (Toulmin, 1958), the same as the German dataset we used for the transfer learning task (Schaller et al., 2024). Ding et al. (2022) trained a sequence tagging model using the pre-trained Longformer (Beltagy et al., 2020) on PERSUADE, achieving an F1 score of .55. We leverage their framework in our experiments.

3 Data

In our experiments, we work with three different datasets: PERSUADE, MEWS, and DARIUS. In the following, we go into details for each dataset, describe our label mapping as the basis for the transfer learning, and compare the sequencing of argumentative elements in each dataset.

EN-L1 The PERSUADE corpus (Crossley et al., 2022) encompasses a collection of 26,000 argumentative essays authored by students in grades 6-12 within the United States, mostly English native speakers. Expertly annotated, these essays feature seven categories of argumentative elements: *lead, position, claim, counterclaim, rebuttal, evidence,* and *concluding statement*. The quality of annotations is evaluated using F1 score reaching an inter-rater agreement (IAA)[2] of 0.73.

EN-L2 The MEWS corpus (Rupp et al., 2019) comprises 9,628 essays written by English-as-a-foreign-language learners in Switzerland and Germany. For this study on transfer learning across L1 and L2 context, we randomly drew and annotated a subset of 110 essays responding to the *Television Advertising* (AD) prompt and 100 essays addressing the *Teachers Ability* (TE) prompt [3]. In terms of writing tasks, these two prompts are close to those in EN-L1 because they are independent writing tasks. These essays were annotated following the same schema as EN-L1, achieving an IAA of F1 = 0.52.

DE DARIUS (Schaller et al., 2024) is a corpus comprising 2,521 texts from the "Energy" prompt

[1] https://www.kaggle.com/c/feedback-prize-2021

[2] The calculation of IAA takes an annotation as a true positive when it is identified by two annotators with over 50% overlap in both directions. Elements identified exclusively by the first annotator are considered false negatives, whereas those only recognized by the second annotator are deemed false positives.

[3] Detailed writing instructions are available on Page 13 and Page 34 at https://www.ets.org/pdfs/toefl/toefl-ibt-writing-practice-sets-large-print.pdf

and 2,517 from the “Automotive" prompt, which are written by German high school students. Similar to the datasets above, this dataset also has an annotation of argumentative elements (with different names, see details in Section 3.2). The IAA among different layers of annotation ranges between 0.57 and 0.98.

The performances of transfer learning on **DE** dataset can be influenced by both language and educational contexts. To keep them apart, we translate it into English as the dataset **DE-MT**, using DeepL Pro[4]. Applying experiments on this dataset would help us distinguish between the impact of the writing task migration and the language transition during transfer learning.

3.1 Dataset Comparison

Table 1 shows the descriptive statistics of the three datasets. We see that EN-L1 and EN-L2 have a comparable length in terms of the average number of sentences (21.25 and 20.56 respectively), whereas the German texts are significantly shorter (9.53). However, this difference does not originate from language, since it is almost the same as the translated data DE-MT (9.81). Instead, this large difference may be attributed to the nature of the writing tasks. As emphasized above, the writing prompts of EN-L1 and L2 are similar, requiring students to produce independent argumentative essays. In contrast, the DE dataset employs integrated writing prompts, potentially leading to shorter, more concise responses.

This point can be also observed in the average number of words per essay, where the DE dataset has the smallest amount (149.89). The EN-L1 dataset leads with 402.31, followed by EN-L2 (349.68). The EN-L2 dataset, despite having a larger average number of sentences, exhibits fewer average words per essay. This observation suggests that L2 writers tend to compose shorter sentences, aligning with findings from the prior study (Burrough-Boenisch, 2002).

Dataset	#Essays	ϕ#Sentences	ϕ#Words
EN-L1	26,000	20.56	402.31
EN-L2	210	21.25	349.68
DE	5038	9.53	149.89
DE-MT	5038	9.81	163.13

Table 1: Descriptive statistics of datasets.

3.2 Label Mapping

For a consistent annotation mapping across diverse datasets, we adopt a streamlined label-set inspired by Ding et al. (2024) for the EN-L1 and EN-L2 datasets. Specifically, we employ the labels *lead, position, claim, evidence*, and *conclusion*, by merging the labels *counterclaim* and *rebuttal* into a single label *claim*. The labels are defined as follows.

- *Lead*: an introduction to grab the reader's attention and point toward the position.
- *Position*: an opinion on the main question.
- *Claim*: a claim that supports the position, refutes another claim or gives an opposing reason to the position.
- *Evidence*: ideas or examples that support claims.
- *Conclusion*: a concluding statement that restates the claims

The DE dataset has a four-layer annotation schema. On the *Content Zone* layer, *introduction, main part* and *conclusion* are labeled to delineate the text's framing and structure. On the *Major Claim* layer, sentences referring to the author's final position on the given topic are labeled as *major claims*. While the *Argument* layer, focused on argument quality, is less relevant to our argument mining study, the layer of *Toulmin's Argumentation Pattern (TAP)* is directly pertinent. This layer aligns with the argument schema in EN-L1 and EN-L2, encompassing the annotated elements:

- *Claim*: an assertion that characterizes the position taken.
- *Data*: fact that provides the basis for a claim.
- *Warrant*: an aspect that explains to what extent data supports a claim.
- *Rebuttal*: an objection to a presented data and/or warrant.

Based on the above definitions, the mapping detailed in Table 2 is established to facilitate our transfer learning approach. Firstly, these five types of argumentative elements in three datasets can be compared in the following analysis. Secondly, we can train an argument mining model detecting these elements on the EN-L1 dataset and test its transferability on the other datasets (zero-shot transfer in Section 4). With the label mapping, the essays in EN-L2 and DE can also be added gradually to fine-tune this model for potentially better performance (learning curve study in Section 5). In the

[4] https://www.deepl.com/pro?cta=header-pro

following, we refer to the mapped labels by their names in the English datasets, i.e. *lead, position, claim, evidence* and *conclusion*.

EN-L1 and EN-L2	Annotation Layer in DE	Label in DE
Lead	Content Zone	introduction
Position	Major Claim	major claim
Claim	TAP	claim or rebuttal
Evidence	TAP	data or warrant
Conclusion	Content Zone	conclusion

Table 2: Label mapping of three datasets.

3.3 Analysis - Label Distribution and Length

Figure 2 visualizes the distribution and the average length (in the number of words) of five types of argumentative elements in the respective dataset.

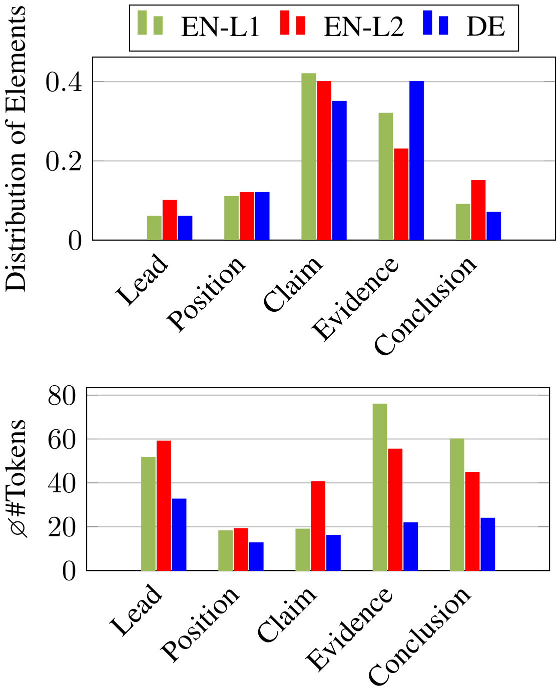


Figure 2: Distribution (upper) and average number of tokens (down) of argumentative elements in three datasets.

The distribution of various argumentative elements is generally comparable among three datasets, with both *claim* and *evidence* emerging as the dominant major classes across all datasets. For the average length in general, DE has the least number of words in all the elements. This again corresponds to our previous analysis of text length, that German argumentative essays tend to be briefer and might have originated from its integrated writing prompts.

The *claim* is most frequent in the EN-L1 dataset, followed closely by EN-L2, while DE exhibits a slightly lower frequency. This suggests a consistent emphasis on presenting central arguments across both English datasets. However, EN-2 stands out with the longest average length for *claim*, suggesting that argumentative essays written by second-language learners may provide more detailed or elaborate claims for central positions compared to native speakers and German writers.

DE exhibits the highest frequency of *evidence* labels among the three datasets, indicating a relatively higher occurrence of supporting details in argumentative essays compared to EN-L1 and EN-L2. However, DE also has the shortest average length for this label. It indicates that although EN-L2 has a higher frequency of evidence, the individual instances are shorter. It could also suggest the possibility of multiple spans or fragmented annotations for longer evidence segments.

EN-L2 stands out with the highest frequency of *lead* and *conclusion* labels, implying emphasis at the beginning and end of essays by non-native English writers. In contrast, native writers (EN-L1 and DE) display lower percentages for these labels. Especially for DE, it exhibits both the lowest frequency and the shortest average length of *conclusion*, suggesting a brief concluding style in German essays.

3.4 Analysis - Label Transition

To examine the structure of essays in datasets, we visualize the argumentation flow as transition graphs where argumentative elements correspond to states and arrows mark the transitions from one element to another annotated with the transition probability (Figure 3). We add two states ‘START’ and ‘END’, indicating the beginning and end of an essay. For a clearer illustration, all transition arrows with probabilities below 0.2 are omitted.

EN-L1 essays (left subfigure) predominantly start with a *lead* and follow with a *position*. Subsequently, the transition to *claim* is most likely, and from there, essays often transition to another *claim* or an *evidence*. Finally, almost all the essays end with *conclusion*. This style is influenced by the five-paragraph essay model, which is the most frequently taught form of writing in classrooms in the US (Campbell, 2014). It usually consists of one introductory paragraph, three body paragraphs for support, and one concluding paragraph.

Similar to EN-L1, EN-L2 essays (sub-figure in the middle) also start with a *lead* predominantly. However, the *lead* is no longer followed directly by the *position*, but by *claims*. Instead, the *position*

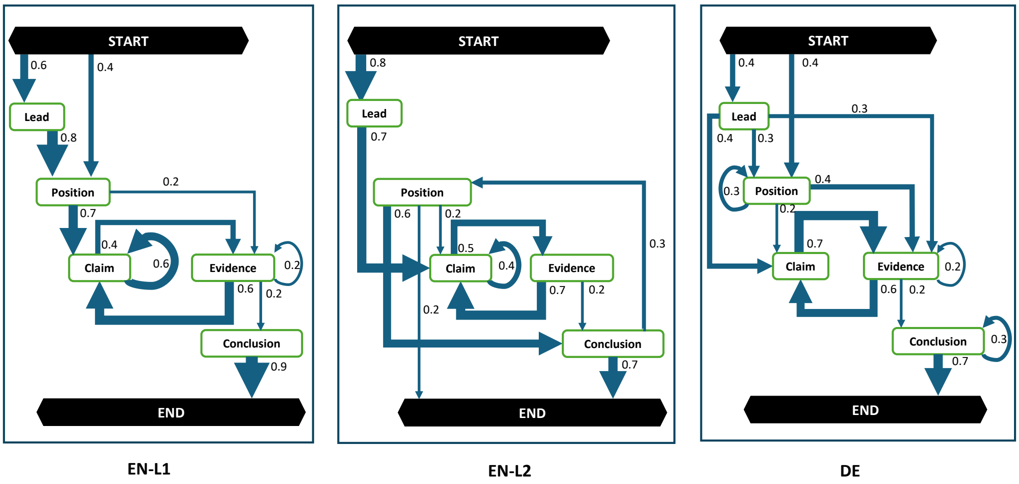


Figure 3: Transitions of different elements in the essays from three datasets.

can be mostly found at the end of the essays, which is illustrated by the 0.6 probability of transitioning from *position* to *conclusion*, as well as the 0.3 probability of transitioning backward from *conclusion* to *position*. By delving deeply into the teaching guidance of argumentative writing in Germany, we found a possible reason for this phenomenon: when it comes to stating a position in argumentative writing, German students are encouraged to state their own opinion at the end for a balanced discussion (Becker-Mrotzek et al., 2010).

The structuring style in German essays (right sub-figure) is more diverse. Firstly, almost the same amount of essays start with a *lead* or a *position*, which aligns with the suggestion in the earlier study that arguments in German have a higher level of directness (Tannen, 1998). In other words, German writers tend to jump straight into the position instead of introducing the topic first with a lead. Besides *claim* and *position*, 40% *lead* was directly followed by the *evidence*. Unlike the English datasets, the *claims* in DE are rarely followed by another claim but dominantly followed by an *evidence*. This discrepancy can be attributed to the integrated writing task in DE, which imposes a greater demand on students to integrate evidence from sources into their writing (Cumming et al., 2005). At last, we notice that more self-transitions in DE (30% *conclusions*, 20% *evidence* and 30% *positions*), which may not be an inherent property of the essays but rather an annotation artifact based on a high granularity.

4 Study 1: Zero Shot Transfer

For our first study, we adopt the sequence tagging architecture developed by Ding et al. (2022), which pre-processes the annotated training data into tokens with Inside-Outside-Beginning (IOB) tags and uses them as the input to the pretrained Longformer model (Beltagy et al., 2020) for token classification. We trained two models on 90% of EN-L1 data with the *Longformer*[5] to transfer on EN-L2 data and its multi-lingual variation *XLM-R Longformer*[6] for the DE data. After 10 epochs of training with a maximal length of 1024 tokens, the IOB tags of tokens are post-processed into predictions for different argumentative elements.

Following the same schema as for the IAA evaluation, we evaluate our results also through the F1 score: all gold standards and predictions for a given argumentative element are compared. If the overlap between the gold standard and prediction in both directions is higher or equal to 0.5, the prediction is considered a true positive. If multiple matches exist, the match with the highest is taken. Any unmatched ground standards are false negatives and any unmatched predictions are false positives.

4.1 EN-L1 to EN-L2

Table 3 shows the performance of the argument mining model tested on EN-L1 and two different prompts on EN-L2. Overall, the transfer performance of the model achieves an F1 score of 0.56 and 0.42 on EN-L2 dataset, which is only a slight drop from the performance on EN-L1, indicating its effectiveness in extracting argumentative elements from essays written by both native and non-native English speakers. The model demonstrates the best

[5] https://huggingface.co/allenai/longformer-base-4096

[6] https://huggingface.co/markussagen/xlm-roberta-longformer-base-4096

proficiency in detecting *lead* elements across all datasets. A possible explanation is that this element is often found at the beginning of essays and therefore easy to find. The transferability of the model tested on the prompt AD is better than TE, implying the prompt similarity between AD and EN-L1 is higher than TE and EN-L1.

	Test Data		
	EN-L1	**EN-L2 AD**	**EN-L2 TE**
Lead	.76	.79	.63
Position	.61	.57	.28
Claim	.44	.41	.28
Evidence	.69	.49	.39
Conclusion	.78	.52	.50
Overall	.66	.56	.42

Table 3: Zero shot transfer from EN-L1 to EN-L2 with English pretrained Transformer

We examine the typical misclassification in the two prompts together. The confusion matrix in Table 4 illustrates that most of the confusion arises between a label and no assigned span, suggesting challenges in accurately delineating argumentation unit boundaries. More specifically, a gold argument is often divided into multiple predicted spans or vice versa. This issue results in numerous spans lacking a counterpart with significant overlap. Among the instances of actual confusion between two labels, we noted a common misclassification of *evidence* being incorrectly labeled as *claims*.

	Lead	**Position**	**Claim**	**Evidence**	**Conclusion**	**None**
Lead	100	4	6	14	0	86
Position	4	61	4	4	6	142
Claim	7	8	147	120	11	447
Evidence	5	2	14	157	14	205
Conclusion	0	11	6	15	64	158
None	72	79	277	284	89	N.A.

Table 4: Confusion matrix between gold standards (columns) and predictions (rows) of EN-L2.

4.2 EN-L1 to DE

Table 5 shows the result of zero-shot transfer learning from EN-L1 to DE and DE-MT. We first notice that on the same test dataset of EN-L1 the performance decreased by changing the pretrained Longformer into its multi-lingual version XLM-R Longformer. Especially for the label *position*, the F1 score dropped from .61 to .29. These results align with earlier studies, showing multilingual models have worse performance than their monolingual counterparts on certain downstream tasks (Conneau et al., 2020).

The transfer performance to DE is not as good as EN-L2, as evidenced by the lower F1 scores across all labels. However, the model's performance decline is not solely attributable to language differences between English and German, as even the machine-translated German dataset (DE-MT) exhibits similar performance. The F1 scores for *claim* and *evidence* are particularly low across both the DE and DE-MT datasets. This poor performance is likely influenced by the differences observed in the distribution and length of these elements in the integrated tasks, as discussed in Section 3.3.

	Test Data				
		Energy		**Automotive**	
	EN-L1	**DE**	**DE-MT**	**DE**	**DE-MT**
Lead	.73	.61	.63	.59	.62
Position	.29	.28	.35	.32	.32
Claim	.38	.15	.17	.14	.16
Evidence	.65	.28	.29	.27	.30
Conclusion	.74	.48	.50	.48	.48
Overall	.61	.36	.38	.36	.38

Table 5: Zero shot transfer from EN-L1 to DE and DE-MT with multi-lingual Transformer

The confusion matrix in Table 6 shows the same pattern as Table 4. Besides the majority of confusion occurring between a label and no assigned span, *claim* and *evidence* are often wrongly switched. When comparing the number of unmatched gold standard labels (7036) with that of unmatched predicted labels (25779), we see the model tends to assign a label rather than not assign anything.

	Lead	**Position**	**Claim**	**Evidence**	**Conclusion**	**None**
Lead	1102	357	76	58	0	630
Position	18	1147	285	133	41	3079
Claim	44	217	1319	1114	135	10276
Evidence	29	61	1028	3301	66	10702
Conclusion	0	445	74	128	1041	1092
None	261	614	2073	3813	275	N.A.

Table 6: Confusion matrix between gold standards (columns) and predictions (rows) of DE.

In summary, while the performance of our argument mining model does not match that achieved on the source dataset (EN-L1), considering it does not see any data from the target domain during training, it performs reasonably well in different educational contexts and languages (EN-L2 and DE). This highlights the potential of the generalization capability of this model.

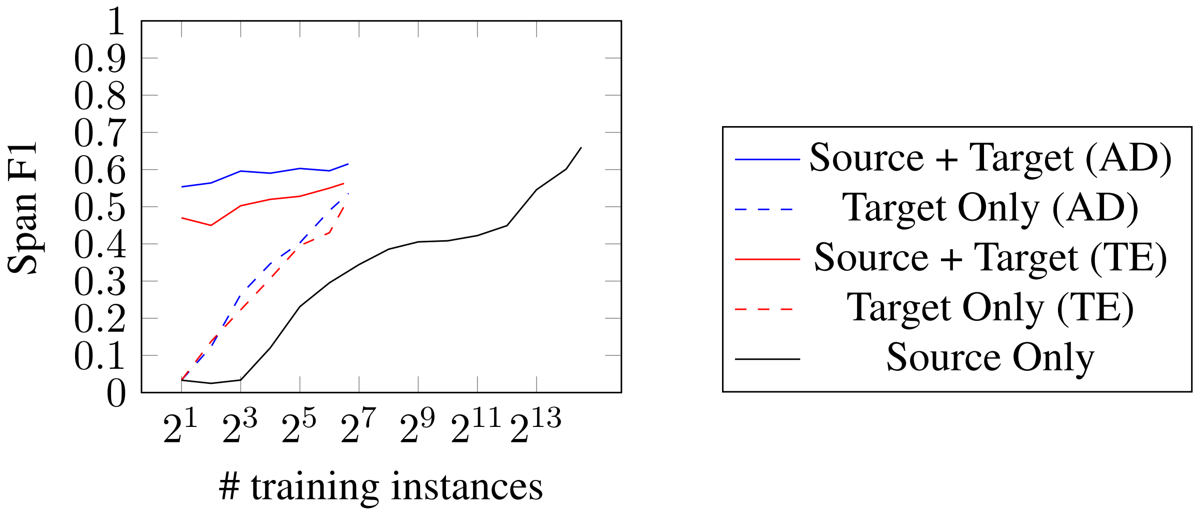


Figure 4: Learning curve EN-L2 and EN-L1

5 Study 2: Learning Curve

After having established zero-shot transfer performance, we investigate the potential of using a small amount of target domain training data to improve the performance of our argument mining model on the target test dataset. This process involves fine-tuning the model trained on EN-L1 data using a portion of data from the target domain (EN-L2 and DE), allowing it to adapt its representations of the features in L2 and German argumentative essays.

However, it is important to note that fine-tuning requires access to labeled data from the target domain. In a practical application scenario, when a teacher wants to fine-tune such a model for a new educational context or language, it is important to know how much data needs to be labeled, since human annotation effort is often a crucial factor.

Therefore, we perform a series of learning curve experiments, in which we systematically vary the amount of training data from target datasets.

5.1 EN-L1 to EN-L2

Since EN-L2 only has 210 labeled data, we use the ten-fold cross-validation data splitting and report the average performance. On each training data set, we fine-tune the model from zero-shot transfer for 10 epochs. In comparison to the fine-tuning (**Source+Target**), we also trained the Longformer from the beginning only using these training data from EN-L2 (**Target Only**).

Figure 4 plots the amount of training data from the target domain on the x-axis and the model performance (F1) on the y-axis. Both Source+Target curves start with relatively high F1 scores but exhibit slow growth as the number of training instances increases. In contrast, the "Target Only" curves demonstrate faster growth with increasing training instances. However, despite this rapid improvement, these lines do not achieve the same level of performance as the "Source + Target" scenarios. This indicates that the current amount of labeled data in EN-L2 is insufficient to match the performance achieved by incorporating knowledge from EN-L1. Therefore, the transfer learning strategy is necessary for the limited labeled data in the target domain.

To estimate the amount of data needed for labeling, the "Source Only" curve provides a reference. This curve represents the scenario where the model is trained solely on data from EN-L1. As the number of training instances from the target domain increases, the model performance on the target task is expected to approach the upper bound at F1=.66 with 23,400 labeled training data instances.

5.2 EN-L1 to DE

Figure 5 shows the learning curves of DE and DE-MT datasets. Same to Figure 4, all the Source + Target curves start at a relatively high-performance level but exhibit a slower rate of improvement. Unfortunately, the gap between them and Target Only lines can be quickly narrowed. This implies that in educational context transfer, such as transitioning from independent to integrated tasks, better performance can be attained by training the model from scratch using an adequate amount of labeled data from the target domain.

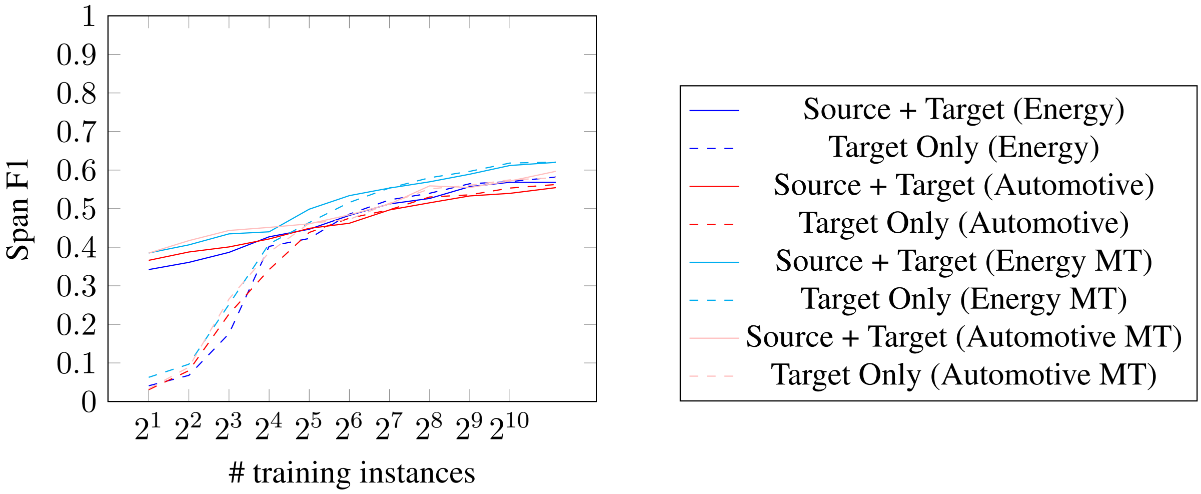


Figure 5: Learning curve DE and DE-MT

Regarding language transfer, the performance achieved through machine translation (MT) is found to be very close to that of the original source. As a result, there appears to be no significant benefit in using machine-translated data for training purposes.

6 Conclusion

This paper explores the transferability of argument-mining models across different educational contexts and languages. Through comprehensive analyses of various datasets, including those authored by native English speakers (EN-L1), English as a foreign language learners (EN-L2), and native German writers (DE), as well as machine-translated German essays (DE-MT), we answer RQ 1 and show their differences in linguistic structures and argumentation styles.

Our experimental studies designed for RQ 2 reveal that, under zero-shot conditions, models trained on EN-L1 demonstrate satisfactory performance when directly applied to EN-L2/DE datasets. However, fine-tuning the model on target domain data does not increase the performance significantly, highlighting the challenges of transfer learning across different educational contexts and languages. Notably, as the answer for RQ 3, machine translation does not significantly enhance performance, indicating that differences in dataset characteristics stem less from language disparities, but more from distinct educational contexts.

7 Limitations

This study showed the transferability of argument mining models for the English-German language pair on three specific corpora. Whether a transfer works equally well for languages phylogenetically further from the source language and potentially less well-covered in pretrained multilingual transformer models remains an open question.

Acknowledgements

This work was partially conducted at “CATALPA - Center of Advanced Technology for Assisted Learning and Predictive Analytics” of the FernUniversität in Hagen, Germany.

References

Tariq Alhindi and Debanjan Ghosh. 2021. “sharks are not the threat humans are”: Argument component segmentation in school student essays. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 210–222.

Michael Becker-Mrotzek, Frank Schneider, and Klaus Tetling. 2010. Argumentierendes schreiben–lehren und lernen. 17:2012.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

J Burrough-Boenisch. 2002. *Culture and conventions: writing and reading Dutch scientific English*. Ph.D. thesis, Utrecht: LOT.

Kimberly Hill Campbell. 2014. Beyond the five-paragraph essay. *Educational Leadership*, 71(7):60–65.

Wei-Fan Chen, Mei-Hua Chen, Garima Mudgal, and Henning Wachsmuth. 2022. Analyzing culture-specific argument structures in learner essays. In *Proceedings of the 9th Workshop on Argument Mining*, pages 51–61.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

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 (PERSUADE) corpus 1.0. *Assessing Writing*, 54:100667.

Alister Cumming, Robert Kantor, Kyoko Baba, Usman Erdosy, Keanre Eouanzoui, and Mark James. 2005. Differences in written discourse in independent and integrated prototype tasks for next generation toefl. *Assessing Writing*, 10(1):5–43.

Joanne Devine, Kevin Railey, and Philip Boshoff. 1993. The implications of cognitive models in l1 and l2 writing. *Journal of second language writing*, 2(3):203–225.

Yuning Ding, Marie Bexte, and Andrea Horbach. 2022. Don't drop the topic-the role of the prompt in argument identification in student writing. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 124–133.

Yuning Ding, Omid Kashefi, Swapna Somasundaran, and Andrea Horbach. 2024. When Argumentation Meets Cohesion: Enhancing Automatic Essay Scoring. Accepted for LREC-COLING 2024.

Steffen Eger, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. 2018. Cross-lingual argumentation mining: Machine translation (and a bit of projection) is all you need! In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 831–844.

Johanna Fleckenstein, Lucas W Liebenow, and Jennifer Meyer. 2023. Automated feedback and writing: A multi-level meta-analysis of effects on students' performance. *Frontiers in Artificial Intelligence*, 6.

Steve Graham, Sharlene A Kiuhara, and Meade MacKay. 2020. The effects of writing on learning in science, social studies, and mathematics: A meta-analysis. *Review of Educational Research*, 90(2):179–226.

Liang Guo, Scott A Crossley, and Danielle S McNamara. 2013. Predicting human judgments of essay quality in both integrated and independent second language writing samples: A comparison study. *Assessing Writing*, 18(3):218–238.

Andrea Horbach, Joey Pehlke, Ronja Laarmann-Quante, and Yuning Ding. 2023. Crosslingual content scoring in five languages using machine-translation and multilingual transformer models. *International Journal of Artificial Intelligence in Education*, pages 1–27.

Andrea Horbach, Sebastian Stennmanns, and Torsten Zesch. 2018. Cross-lingual content scoring. In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 410–419.

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.

John Lawrence and Chris Reed. 2020. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818.

Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Andreas Peldszus and Manfred Stede. 2013. From argument diagrams to argumentation mining in texts: A survey. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 7(1):1–31.

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, Online. Association for Computational Linguistics.

Robert Ridley, Liang He, Xin-yu Dai, Shujian Huang, and Jiajun Chen. 2021. Automated cross-prompt scoring of essay traits. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 13745–13753.

André A Rupp, Jodi M Casabianca, Maleika Krüger, Stefan Keller, and Olaf Köller. 2019. Automated essay scoring at scale: a case study in switzerland and germany. *ETS Research Report Series*, 2019(1):1–23.

T Sanders and Joost Schilperoord. 2006. Text structure as a window on the cognition of writing. *Handbook of writing research*, pages 386–402.

Nils-Jonathan Schaller, Andrea Horbach, Lars Höft, Yuning Ding, Jan L Bahr, Jennifer Meyer, and Thorben Jansen. 2024. Darius: A comprehensive learner corpus for argument mining in german-language essays. Accepted for LREC-COLING 2024.

Christian Stab and Iryna Gurevych. 2014. Annotating argument components and relations in persuasive essays. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1501–1510, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Deborah Tannen. 1998. The argument culture: Moving from debate to dialogue. *New York: Random House Trade.*

Orith Toledo-Ronen, Matan Orbach, Yonatan Bilu, Artem Spector, and Noam Slonim. 2020. Multilingual argument mining: Datasets and analysis. In *Findings of the association for computational linguistics: Emnlp 2020*, pages 303–317.

Stephen E Toulmin. 1958. *The uses of argument*. Cambridge university press.

Sowmya Vajjala and Taraka Rama. 2018. Experiments with universal CEFR classification. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 147–153.

Thiemo Wambsganss, Christina Niklaus, Matthias Söllner, Siegfried Handschuh, and Jan Marco Leimeister. 2020. A corpus for argumentative writing support in German. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 856–869, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. 2016. A survey of transfer learning. *Journal of Big data*, 3:1–40.

Jin Xue, Xiaoyi Tang, and Liyan Zheng. 2021. A hierarchical bert-based transfer learning approach for multidimensional essay scoring. *Ieee Access*, 9:125403–125415.

Anar Yeginbergenova and Rodrigo Agerri. 2023. Cross-lingual argument mining in the medical domain.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76.

Building Robust Content Scoring Models for Student Explanations of Social Justice Science Issues

Allison Bradford and Marcia C. Linn
Berkeley School of Education
University of California, Berkeley
{allison_bradford, mclinn}@berkeley.edu

Kenneth Steimel and Brian Riordan
ETS
{ksteimel, briordan}@ets.org

Abstract

With increased attention to connecting science topics to real-world contexts, like issues of social justice, teachers need support to assess student progress in explaining such issues. In this work, we explore the robustness of NLP-based automatic content scoring models that provide insight into student ability to integrate their science and social justice ideas in two different environmental science contexts. We leverage encoder-only transformer models to capture the degree to which students explain a science phenomenon, understand the intersecting justice issues, and integrate their understanding of science and social justice. We developed models training on data from each of the contexts as well as from a combined dataset. We found that the models developed in one context generate educationally useful scores in the other context. The model trained on the combined dataset performed as well as or better than the models trained on separate datasets in most cases. Comparing human scores with the automated scores using quadratic weighted kappas demonstrate that these models perform above the threshold for use in classrooms.

1 Introduction

This study investigates the robustness of Natural Language Processing (NLP)-based automatic content scoring models that assess secondary school students' ability to integrate their science and social justice ideas to explain social justice science issues (SJSI; Morales-Doyle, 2017) in two different contexts. In particular, we investigate the robustness of content scoring models in terms of their ability to score out-of-distribution responses as we attempt to generalize the models from one SJSI context to another. The contexts are (a) a unit about combustion reactions and asthma caused by exposure to particulate matter pollution and (b) a unit about global climate change and exposure to extreme heat occurring in urban heat islands. In both units, which are also aligned to state science standards, the students explore the racially disparate impacts of the environmental hazard (particulate matter pollution, extreme heat in urban spaces). In the units, students are supported to explore typical disciplinary content and make connections to local justice issues. They answer the *Impacts Item*, explaining whether all people are impacted by the environmental hazard in the same way. We explore the possibility of building a robust domain general model that can be used across multiple SJSI contexts.

The curriculum, assessments, and scoring rubrics were developed by a research practice partnership (RPP) including classroom teachers, computer scientists, and learning scientists guided by the Knowledge Integration pedagogy (KI; Linn and Eylon, 2011). The automatic content scoring models were created to assess the degree to which students connect their understanding of the environmental concepts with understanding of the social justice issues when explaining whether everyone is impacted in the same way. As teachers reformulate their instruction to include social justice perspectives, automatic content scoring models can help teachers by capturing student progress. They are especially valuable for social justice ideas that might be new to science teachers. We investigate the accuracy and robustness of automatic content scoring models that can quickly assess student explanations, particularly when those explanations contain social justice ideas. In this study we ask:

- Can we develop NLP models that accurately capture students' integrated understanding of SJSIs, as measured by human-computer agreement?

- What are the affordances and limitations of combining training datasets from different disciplinary contexts to develop robust automatic content scoring models of SJSI?

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 450–458
June 20, 2024 ©2024 Association for Computational Linguistics

2 Related Work

This study builds on prior research integrating social justice into science curriculum and leveraging AI techniques to score student essays.

2.1 Social Justice Science Issues (SJSI)

In this study, we combined social justice science pedagogy (Morales-Doyle, 2017) with Knowledge Integration (KI) design framework (Linn and Eylon, 2011) to design units featuring SJSI. Centering issues of social justice in science teaching and learning offers promise for preparing students to deal with contemporary science issues. One productive example involves grounding science teaching in local social justice science issues (Morales-Doyle, 2017). Students in this Chicago neighborhood drew attention to the contamination in the soil in the community garden. Introducing SJSIs provides opportunities for students to make sense of issues impacting their own communities and raises issues around inequality and racism (Morales-Doyle et al., 2019). In making sense of such issues, students connect typical science ideas to interpret how an environmental phenomenon impacts their community. This enables them to integrate disciplinary ideas with social justice ideas to explain why the impacts are different across racial and socioeconomic groups.

We also developed an aligned assessment, the *Impacts* item, that requires students to explain an environmental hazard and whether all people are impacted by it the same way. For example, when explaining who is impacted by urban heat islands, a student wrote, *"I don't think all people are impacted by the effects of climate change in the same way. Red areas on the map are 5-20 degrees higher than blue or green areas. Red areas are mostly habited by brown and black people. Red areas have less funding because of segregation and racism. They have less access to government funds and less green areas which help with the decrease in climate change."* To assess student explanations, the KI framework indicates that assessment should focus on the integration of concepts rather than the accuracy of isolated ideas, requiring the development of automatic content scoring models that capture the degree to which students integrate their ideas. As such, we developed an overall KI score rubric (Table 1; Liu et al., 2008; Liu et al., 2016) as well as KI-aligned Disciplinary and Justice subscore rubrics to score training data.

2.2 Automatic Content Scoring

Automatic content scoring can be traced back to early work on the Project Essay Grader (PEG) system which leveraged computers to grade essays and found that a computer rater's score was nearly as highly correlated with human raters' scores as the human raters' scores were with each other (Page, 1966). This work paved the way for Automatic Essay Scoring (AES) models and automatic content scoring. Many advances in AES modeling have resulted in widely used classroom and high stakes assessments. For example the e-rater automated scoring system is used for the Graduate Management Admission Test (GMAT; Burstein, 2003). To score short, student-generated free-text responses such as the *Impacts item* according to a scoring rubric, c-rater has shown promise (Leacock and Chodorow, 2003). C-rater works by determining whether a natural language response is part of the set of correct ideas that could be expressed in response to the prompt. To do so, the model uses a number of natural language processing techniques to normalize a response by attending to sources of variation in expression of the same idea: syntactic variation, morphological variation, pronoun reference, the use of synonyms or similar words, and spelling or grammatical errors.

Recently, researchers working on automatic content scoring for short answer responses have sought to incorporate approaches that have been effective in the realm of AES (e.g. Riordan et al., 2017) like the use of neural architectures (e.g. Zhao et al., 2017) including pre-trained transformer models (e.g. Yang et al., 2020). In particular, we build on the automatic content scoring work of Riordan et al. (2020) which showed that recurrent neural network and encoder-only transformer models performed just as well or better than feature-based models. Riordan et al. (2020) also demonstrated that the encoder-only transformer-based models were more robust to spurious, dataset-specific learning cues when applying scoring rubrics. Thus, we adopt a similar approach of fine-tuning encoder-only transformer models, BERT and SciBERT, to develop short answer scoring models for KI, Disciplinary, and Justice scores.

3 Data and Experimental Design

We developed automatic content scoring models to automatically score the *Impacts* item which is found in several units: Global Climate Change and

KI Score	Criteria	Asthma Example	UHI Example
1	Irrelevant	idk	asifhsdif
2	Vague	Yes, climate change will effect everyone in the whole world.	I think in some ways yes and in some ways no.
3	Partial link: one target idea	Yes, because if you have more freeways or factories where you live you could have more of the effects of incomplete combustion.	No, because there is less greenery, and plants and trees help to keep things cool in urban heat islands.
4	Full link: links two target ideas	People who are lower income are impacted by climate change more than people who aren't because they sometimes have to live closer to factories and other places where there could be harm.	No, some people who for example live in poorer or redlined areas will be more impacted. As those areas don't have as much greenery or architecture that can help with the heat.
5	Full links: links three or more target ideas	NO! Racially oppresed groups are affect more by climate change. These groups are in redlined communities which put near industrial areas which produce green house gases. These greenhouse gas emmisions give you a higher chance to have asthma.	Black and hispanic people who live in poorer residences have less trees and grass nearby, as an effect of redlining, which makes poorer neighborhoods hotter. The rich white neighborhoods are invested in by banks, and have much more trees and grass, making their neighborhood 5-20 degrees cooler.

Table 1: Rubric for KI score with examples from both unit contexts.

Urban Heat Islands (UHI; 9th grade) and Chemical Reactions and Asthma (Asthma; 7th grade). In this section, we describe the item, scoring rubrics, training data, and experimental design. The section that follows details our model development approach.

3.1 Assessment Item and Scoring Rubrics

The *Impacts* item asks students to explain whether all people are impacted by an environmental hazard in the same way. In both unit contexts, students connect their science understanding to the role of race, socioeconomic status, and policies like redlining in their local communities. In the UHI unit, the item prompt elicits ideas about how the Sun transfers energy to different surfaces and how those surfaces contribute to the surrounding temperature. In the Asthma context, the item prompt elicits ideas about how the products of incomplete combustion reactions relate to asthma.

To develop the scoring model for the Impacts item, we first developed a knowledge integration (KI score) rubric (scale 1-5; Liu et al., 2008; Liu et al., 2016) and two subscore rubrics: Disciplinary and Justice (scale 0-2). The KI score measures the overall integration of ideas in the student explanation and is agnostic to the explanation context (see rubric in Table 1). The Disciplinary subscore characterizes how students integrate domain specific target ideas in their explanations. While the rubric structure is the same, the disciplinary target ideas are different in the Asthma and UHI contexts. The Justice subscore characterizes how students integrate target ideas about historical policies and social injustices into their explanations. The justice target ideas are the same in the Asthma and UHI contexts. Target ideas were identified in collaboration with all members of the RPP. A subscore of 0 indicates no mention of target ideas, a subscore of 1 indicates an isolated target idea, and a subscore of 2 indicates the integration of two or more target ideas. All rubrics reward students for linking their ideas and connecting evidence, and do not penalize students for incorrect ideas.

3.2 Training Dataset and Experimental Design

We applied the scoring rubrics to data from previous classroom studies where students responded

Disciplinary Subcore	Criteria	Asthma Example	UHI Example
0	No mention	I think so	Everyone is affected
1	Isolated	Yes, because historical practices like redlining made certain neighborhoods that had poorer air quality be the only neighborhoods available to people of color.	no, some people they are homeless and have it harder when there is no shelter and it's really hot outside. when other people can go inside too and air-conditioned houses.
2	Full link	Many places are redlined and those neighborhoods are usually near freeways and refineries and have poor living conditions. People of color are often the ones forced to live in redlined areas so they deal with the incomplete combustion from the freeways and refineries much more than people who live in an area that is not redlined.	People of color and people in lower-income households are much more likely to experience the effects of a global rising temperature. They are less likely to be able to afford proper air conditioning and to live near green areas, which causes an increased rate of heat-related hospital visits and deaths.

Table 2: Rubric for Justice subscore with examples from both unit contexts.

Disciplinary Subcore	Criteria	Asthma Example	UHI Example
0	No mention	Probably	Yes because everyone lives in the world and global warming affects all parts of the planet.
1	Isolated	Depending on how many Carbon Monoxide and Particulates there are, which is influenced by factories. If you live closer or work in factories, the effect will be much worse	No, because there is less greenery, and plants and trees help to keep things cool in urban heat islands.
2	Full link	Some places have more incomplete combustion, that can make soot and carbon monoxide. This can affect the air quality that people breath in, which causes more cases of asthma or other medical conditions.	Lower-income families and neighborhoods are affected by the lack of trees and greenery to cool down the temperatures. It can affect the residents towards more respiratory diseases, heart problems, or dehydration.

Table 3: Rubric for Disciplinary subscore with examples from both unit contexts.

to the Impacts item. Available data included 1690 responses from the Asthma unit and 548 responses from the UHI unit. The student responses are short essays, typically ranging from 1-3 sentences long. The students represented in the training data are from the 6th-9th grade in schools in a large, Western United States metropolitan area.

To assess reliability of human scoring before building the models, two raters independently applied the rubrics to 5 percent of the data and then calculated Pearson's kappa to measure our interrater reliability. We discussed disagreements and refined the rubrics. We repeated the process until we achieved a kappa > 0.85 for the KI, Disciplinary, and Justice scores. The remaining data was split 50-50 among the two raters and hand scored.

Given the considerably smaller number of responses from the UHI unit, we wondered if data from the Asthma unit context could be used to supplement the data from the UHI unit context to enhance the likelihood of developing a scoring model that performs well (as measured by alignment to human scoring). With this in mind, we established three training datasets: 1) the 548 responses collected in the UHI unit context, 2) the 1690 responses collected in the Asthma unit context, and 3) the combined 2238 responses collected across both unit contexts. Descriptive statistics for KI, Disciplinary, and Justice scores for each of the training datasets can be found in Table 4. To evaluate the effect of the composition of the training dataset, we developed the three scoring models using each training dataset. This resulted in nine total models:

- UHI-trained KI
- UHI-trained Disciplinary
- UHI-trained Justice
- Asthma-trained KI
- Asthma-trained Disciplinary
- Asthma-trained Justice
- Combined-trained KI
- Combined-trained Disciplinary, and
- Combined-trained Justice.

4 Models

4.1 Modeling Approach

The human-scored data in three training datasets were used to train content scoring models for KI, Disciplinary, and Justice scores. The models were based on encoder-only transformer models (in this case, BERT and SciBERT), following prior work (Riordan et al., 2020). The models for KI, Disciplinary, and Justice scores were trained independently, with each score representing the degree of integration for the corresponding aspect of the content of the response. Models were trained on ordinal scores (1-5 for KI, 0-2 for Justice and Disciplinary) using the text in each response. The modeling approach was a standard "instance-based" approach (as opposed to similarity-based approach; c.f. Horbach and Zesch, 2019). While instance-based models may not generalize well across prompts (Horbach and Zesch, 2019), we anticipated that responses generated by UHI and Asthma versions of the *Impacts* item would succeed because many ideas or phrases associated with high level scores are the same in both unit contexts. Ideas that are specific to a particular unit context are unlikely to occur in the other context, minimizing the likelihood that words or phrases associated with a high score from one unit context would be associated with a low score in the other unit context.

We used BERT (Devlin et al., 2019) for the KI and Disciplinary scores and SciBERT for the Justice score (Beltagy et al., 2019). The backbone selection was based upon prior experimentation not reported in this paper. Following standard practice, for all models, during training, a special classification token '[CLS]' was added to the beginning of each input sequence. To make score predictions, the learned representation for the [CLS] token was processed by an additional layer with sigmoid activation, outputting a real-valued score prediction. This real value was mapped back to ordinal scores for making predictions.

During training, learning rates were tuned individually for each model using grid search. Hyperparameter optimization was carried out as follows: We trained using 10-fold cross-validation with an 80-10-10 training/validation/test split. We tuned hyperparameters by training on each train split and evaluating on validation splits. We retained the epoch where best performance was observed and the predictions from that epoch. Then, to select

Training Dataset	Mean	Median	Min	Max	Std Dev
UHI-KI	2.73	3	1	5	0.82
Asthma-KI	2.75	3	1	5	0.78
Combined-KI	2.75	3	1	5	0.79
UHI-Disciplinary	0.79	1	0	2	0.58
Asthma-Disciplinary	0.82	1	0	2	0.61
Combined-Disciplinary	0.81	1	0	2	0.60
UHI-Justice	0.21	0	0	2	0.44
Asthma-Justice	0.17	0	0	2	0.40
Combined-Justice	0.18	0	0	2	0.41

Table 4: Descriptive statistics for KI, Disciplinary, and Justice Scores for each training dataset

the best hyperparameters, we evaluated the performance of the pooled predictions across all folds of the validation sets. We trained final models by training on the combined train and validation sets, using 10-fold cross-validation and using the best-performing hyperparameters from the prior hyperparameter optimization.

4.2 Classroom Testing and Model Evaluation

After developing the models, we performed additional evaluation using a sample from newly collected classroom data. To evaluate the models on new data, we embedded the *Impacts* item at three time points in both the UHI and Asthma units: on a pretest, within the lesson about the SJSI, and on a posttest. Two ninth grade science teachers taught the UHI unit (student N= 95) and one seventh grade science teacher taught Asthma (student N = 56). We selected a balanced sample of 100 responses from each unit to evaluate the models we built. The responses were human scored and scored by each of the models. We used QWK, a measure of agreement for ordinal ratings that ranges from 0 to 1 and accounts for chance agreement (Fleiss and Cohen, 1973), to compare the performance of the scoring models trained on each training dataset.

5 Results and Discussion

5.1 RQ1: Developing a model to capture students' integrated understanding of SJSI

After model development, we evaluated each model (UHI-trained, Asthma-trained, and Combined-trained KI, Justice and Disciplinary scoring models) on 100 student responses from both the Asthma and the UHI units. The test data were collected during classroom testing and not present in the training dataset during model development. The responses were hand scored by the first author and scored by each of the models. We used quadratic weighted kappa as a metric to evaluate model performance (Table 5).

All models developed performed sufficiently well ($QWK \geq 0.70$, rounded normally; Williamson et al., 2012) in the evaluation context that corresponded to the training context, i.e Asthma-trained KI, Disciplinary, and Justice models performed sufficiently well on new data collected from student learning from the Asthma unit. UHI-trained models performed sufficiently well on new data from students learning the UHI unit. The Combined-trained models also performed sufficiently well ($QWK \geq 0.70$, rounded normally; Williamson et al., 2012) for new data collected in both the Asthma and UHI units. These results suggest that we can automatically assess student progress in explaining SJSI.

5.2 RQ2. Affordances and limitations of combining datasets to develop AES models for similar instructional contexts

In most cases, the model built on a larger training dataset performs better, even if the training dataset includes data from a different instructional context. For example, the Combined-trained Disciplinary model performed best for data from both the Asthma ($QWK = 0.9380$) and the UHI units ($QWK = 0.8273$). Additionally, the Asthma-trained models perform better or as well as UHI-trained models for test data from the UHI context. Figures 1 and 2 illustrate the trend that as more data is added to the training dataset, the QWK either remains approximately the same or increases.

An exception to this trend are the models for the Justice score (Figure 3). The Asthma-trained Justice model performs best for data from both

Training Context	Evaluation Context	KI QWK	Disciplinary QWK	Justice QWK
Asthma (N=1690)	Asthma	0.9649	0.9265	0.9323
UHI (N=548)	UHI	0.9071	0.7531	0.6983
Asthma (N=1690)	UHI	0.9137	0.7499	0.8344
UHI (N=548)	Asthma	0.7941	0.5479	0.8177
Combined (N=2238)	Asthma	0.9385	0.9380	0.8785
Combined (N=2238)	UHI	0.9432	0.8273	0.7922

Table 5: Model evaluation results (quadratic weighted kappa, QWK) on the 100 newly collected student responses for models trained on data from the Asthma context, the UHI context, and the Combined dataset.

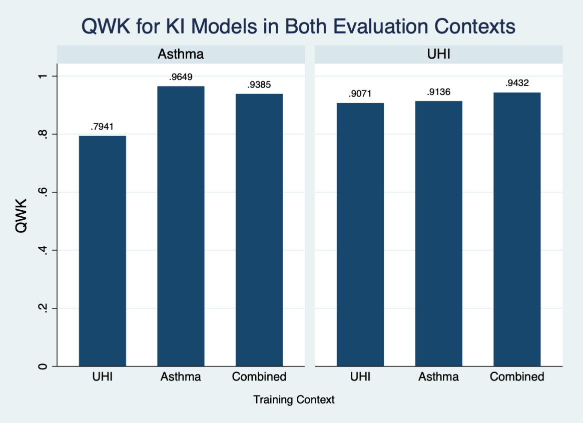


Figure 1: QWK for KI score for each model in both evaluation contexts

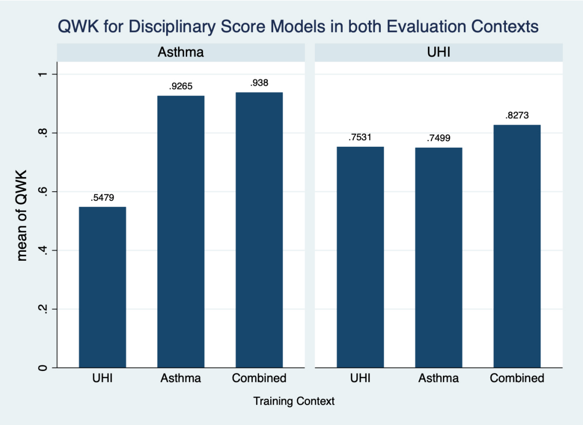


Figure 2: QWK for Disciplinary score for each model in both evaluation contexts

the Asthma unit and the UHI unit compared to the Combined-trained Justice model, even though it was trained using a smaller dataset and does not contain responses from the UHI unit. Of the 100 UHI test responses, there were six responses where the Asthma-trained Justice model accurately scored the response and the Combined-trained Justice model did not accurately score the response. In each of these responses, the Combined-trained model scored the response lower than the human rater. Four of these six responses were scored at a level 2, the highest score, by the human rater and Asthma-trained model and at a level 1 by the Combined-trained model. For example, the student explanation, "No, people are affected differently by climate change. The reasons behind it are also racially driven, as those who are affected more are likely to be people of color due to redlining and the zoning of housing" was accurately given a Justice score of 2 by the Asthma-trained Justice model and given a score of 1 by the Combined-trained Justice model. The ideas about people of color being more impacted due to historical redlining and housing policies contained in this responses are well represented in the Asthma training dataset.

With this in mind, a possible explanation for the difference in performance is that the Asthma dataset has more responses and more consistent representation of the target justice ideas. As such, it might be reasonable to expect it to perform best. Further, the justice context requires real world knowledge which is a difficult task for transformer models. Additionally, the average Justice score across the 100 UHI test responses was 0.66, while the average of the Justice scores predicted by the Asthma-trained models was 0.61, the average of the Justice scores predicted by the UHI-trained models was 0.45 and the the average of the Justice score predicted by the Combined-trained models was 0.52. The lower average predicted scores

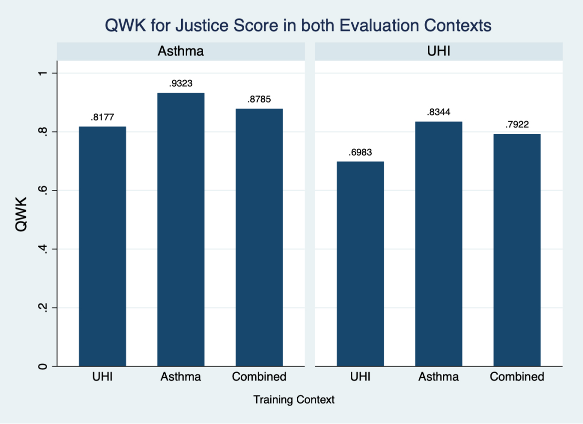


Figure 3: QWK for Justice score for each model in both evaluation contexts

from the UHI-trained and Combined-trained models might indicate that the justice ideas represented in the UHI training dataset are not well-aligned to the justice ideas expressed in the newly collected UHI test set.

Despite some reductions in performance, the Combined-trained KI, Justice, and Disciplinary models all perform well enough to be used in classrooms (Williamson et al., 2012). For the UHI instructional context, where training data was limited during model development, the combined model enhances performance suggesting the promise of the modeling approach for developing a model for in multiple instructional contexts.

6 Conclusions and Next Steps

This study investigates the robustness of pedagogically aligned automatic content scoring models trained for one SJSI context when used for a different SJSI context and of the model trained on multiple SJSI. We found that the models are robust across these contexts. Models developed in one context generate educationally useful scores in the other context. The model trained on the combined dataset is as good or better than the models trained on separate datasets in most cases. These findings underscore the value of using classroom data to fine-tune encoder-only transformer models using a pedagogically-grounded scoring rubric. In particular, the models were robust for scoring student responses for knowledge integration. They also demonstrate the potential for using "instance-based" models across contexts when it is unlikely that words or phrases associated with a high score from one context would be associated with a low score in another context.

Results demonstrate that these models are above threshold for use in classrooms to give students adaptive, personalized guidance based on their essay scores. They can also be used to synthesize classroom data for teachers in real time. Thus, the automatic content scoring generates KI scores, Disciplinary scores, and Justice scores that can be displayed in class-level histograms along with illustrative student responses to help teachers monitor class progress. These results suggest promise for generalizing models across similar contexts, increasing the efficiency of design of automatic content scoring models for adaptive instructional materials.

Next steps include validating the educational value of the models in classroom settings. We plan to engage the RPP in designing and testing adaptive guidance informed by KI pedagogy for each of the automatically generated scores. In addition classroom observations and interviews with teachers are needed to understand how the scores generated by the models align with teachers' assessment of student explanations of SJSIs and how access to student scores from the models shapes their instruction.

7 Limitations

The findings of the work are limited by the nature of our experimental approach. We tested models based on the data available as opposed to systematically testing training dataset size. Further, across all training data sets, the data are imbalanced with an over representation of low Justice scores. These limitations are common constraints when working with data generated in real K-12 classroom contexts.

8 Acknowledgements

We thank Dr. Zachary Pardos for his constructive feedback on an earlier version of this work and the WISE-TELS research group for their feedback. Kenneth Steimel can now be reached at ksteimel@outlook.com and Brian Riordan can be reached at bwriordan@gmail.com.

References

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.

Jill Burstein. 2003. The e-rater® scoring engine: Automated essay scoring with natural language processing.

Jacob Devlin, Kenton Chang, Ming-Wei Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Joseph L Fleiss and Jacob Cohen. 1973. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement*, 33(3):613–619.

Andrea Horbach and Torsten Zesch. 2019. The influence of variance in learner answers on automatic content scoring. In *Frontiers in education*, volume 4, page 28. Frontiers Media SA.

Claudia Leacock and Martin Chodorow. 2003. C-rater: Automated scoring of short-answer questions. *Computers and the Humanities*, 37:389–405.

Marcia C Linn and Bat-Sheva Eylon. 2011. *Science learning and instruction: Taking advantage of technology to promote knowledge integration*. Routledge.

Ou Lydia Liu, Hee-Sun Lee, Carolyn Hofstetter, and Marcia C Linn. 2008. Assessing knowledge integration in science: Construct, measures, and evidence. *Educational Assessment*, 13(1):33–55.

Ou Lydia Liu, Joseph A Rios, Michael Heilman, Libby Gerard, and Marcia C Linn. 2016. Validation of automated scoring of science assessments. *Journal of Research in Science Teaching*, 53(2):215–233.

Daniel Morales-Doyle. 2017. Justice-centered science pedagogy: A catalyst for academic achievement and social transformation. *Science Education*, 101(6):1034–1060.

Daniel Morales-Doyle, Tiffany Childress Price, and Mindy J Chappell. 2019. Chemicals are contaminants too: Teaching appreciation and critique of science in the era of next generation science standards (ngss). *Science Education*, 103(6):1347–1366.

Ellis B Page. 1966. The imminence of... grading essays by computer. *The Phi Delta Kappan*, 47(5):238–243.

Brian Riordan, Sarah Bichler, Allison Bradford, Jennifer King Chen, Korah Wiley, Libby Gerard, and Marcia C Linn. 2020. An empirical investigation of neural methods for content scoring of science explanations. In *Proceedings of the fifteenth workshop on innovative use of NLP for building educational applications*, pages 135–144.

Brian Riordan, Andrea Horbach, Aoife Cahill, Torsten Zesch, and Chungmin 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.

David M Williamson, Xiaoming Xi, and F Jay Breyer. 2012. A framework for evaluation and use of automated scoring. *Educational measurement: issues and practice*, 31(1):2–13.

Ruosong Yang, Jiannong Cao, Zhiyuan Wen, Youzheng Wu, and Xiaodong He. 2020. Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1560–1569.

Siyuan Zhao, Yaqiong Zhang, Xiaolu Xiong, Anthony Botelho, and Neil Heffernan. 2017. A memory-augmented neural model for automated grading. In *Proceedings of the fourth (2017) ACM conference on learning@ scale*, pages 189–192.

From Miscue to Evidence of Difficulty: Analysis of Automatically Detected Miscues in Oral Reading for Feedback Potential

Beata Beigman Klebanov, Michael Suhan, Zuowei Wang, Tenaha O'Reilly
ETS, Princeton NJ, USA
{bbeigmanklebanov,msuhan,zwang,toreilly}@ets.org

Abstract

This research is situated in the space between an existing NLP capability and its use(s) in an educational context. We analyze oral reading data collected with a deployed automated speech analysis software and consider how the results of automated speech analysis can be interpreted and used to inform the ideation and design of a new feature – feedback to learners and teachers. Our analysis shows how the details of the system's performance and the details of the context of use both significantly impact the ideation process.

1 Introduction

Reading a text fluently – accurately and with a good speed – is evidence of development of foundational reading skills, such as decoding and word recognition (Sabatini et al., 2019). Research suggests that the development of oral reading fluency is an essential bridge from decoding to comprehension of text (Pikulski and Chard, 2005). Instructional approaches to foster fluency include modeling fluent reading to the developing reader; repeated reading (re-reading passages multiple times); and engaging students in wide independent reading (Ardoin et al., 2016; Hudson et al., 2020; Pikulski and Chard, 2005; Wexler et al., 2008). Extensive reading was also shown to support fluency development in students of English as a foreign language (Huffman, 2014; Suk, 2017).

Given the importance of fluency for reading development, we built Relay Reader™, an app[1] where readers can practice by taking turns reading out loud from full-length stories with skilled audiobook narrators (Madnani et al., 2019). The narrator reads a passage while the user follows in the text; then the user reads the next passage aloud, and so on. Users can set the word counts of their own and narrator turns between 70 and 200 words.[2] The app has been available since 2020 to readers-in-the-wild, initially with one story (an English translation of Collodi's *The Adventures of Pinocchio*) and gradually expanding to 26 stories, from a 460-word fable to a 120K-word novel. In parallel, the app has been used for independent reading ('Drop Everything And Read') in school and summer camp contexts.

During the development of the app, we conducted a needs assessment with teachers which showed that obtaining estimates of students' oral reading fluency and accuracy was the top priority, followed by being able to see students' specific difficulties in reading through miscue analysis (Kannan et al., 2019). Accordingly, a speech analysis system was developed and is currently used to provide fluency information to teachers. Fluency is measured as words read correctly per minute; hence the system transcribes the audio and compares to the passage text in order to provide fluency estimates. As a byproduct, the system produces an alignment between the transcript and the passage from which the miscues can be easily recovered.

The goal of this study is to explore the potential of using these miscues for feedback, guided by the following **research questions**: (1) What is the extent of miscues in the data? (2) How are miscues distributed in reading passages? (3) How does the extended reading context come into play? (4) How reliable is miscue detection?

The main contribution of our work is the exploration of the space between a deployed NLP capability and its use case. We show how the analysis of the data collected through the system can support ideation of using the system in a new way – for feedback, specifically regarding frequency and content of such feedback. More generally, in the context of using NLP for building educational applications, we zoom in on the process of **ideating**

[1] The app is available freely on https://relayreader.org/.

[2] These are approximate since turn transitions happen on paragraph breaks only.

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 459–469
June 20, 2024 ©2024 Association for Computational Linguistics

a new feature in an existing ecosystem, and show how the analysis of existing data can inform the ideation process.

2 Related work

Research on automated speech recognition (ASR) for young readers suggests that misreadings and slow reading constitute significant challenges (Gelin et al., 2021; Wu et al., 2019); focus on subword units (Hagen et al., 2007) and data augmentation with synthetically generated mistakes (Gelin et al., 2023) are some of the approaches proposed to improve identification of misreadings. The technical challenges notwithstanding, ASR has long been used for feedback in automated reading tutors. The Reading Tutor from Project LISTEN, an influential early system that entered classrooms in the 1990s, displayed the text one sentence at a time. As the student read, the system interrupted if a word was read incorrectly and not self-corrected by underlining the incorrect word and occasionally "coughing" to get the student's attention (Mostow and Aist, 1999). Lalilo is another reading tutor for early elementary students. Students record themselves reading a word, phrase, or sentence; their recording is played back, followed by a fluent model of the sentence. The reader gets feedback when the system is confident that it was correct ('Perfect') or incorrect ('Try again'); if uncertain, the student is asked whether their recording matched the fluent one and is encouraged with 'Good job!' (Hembise et al., 2021). BookBuddy is a chat bot that converses with young readers about the story they are reading by answering their questions, quizzing them, and automatically evaluating their spoken answers (Ruan et al., 2019). The Charlesbridge Reading Fluency program 'listens' as a student is reading, and when a child misreads or struggles with a word, the machine models it and asks the child to repeat it and continue reading; problem words are marked in a separate report for review and practice (Adams, 2013). The virtual reading tutor Marni tracks the student while reading aloud by moving the cursor to each word as it is spoken (Cole et al., 2007). A reading tutor for Dutch supports reading individual words, word lists, and short stories; for the latter, the student is asked to reread the sentences where they read incorrectly one or more words, as detected by ASR software (Bai et al., 2020).

In general, prior work on automated fluency support tends to focus on very young learners (K-2) and on an early stage of fluency development, using words, sentences, or, at most, very short stories, and on helping the student get every item right. In contrast, Relay Reader is targeting a *more advanced* stage of fluency, with a focus on *immersive extended* reading. In this context, it may not be very important to get every word right, especially if it comes at the cost of breaking the flow of reading. Still, detailed speech analysis data similar to that available in reading tutors can be obtained and can therefore be used for stakeholder feedback. This work is a preliminary investigation towards designing miscue-based feedback appropriate to the extended reading application.

3 Data

The data for this study come from users-in-the-wild and from study participants in school and summer programs. Users-in-the-wild may choose to respond to a few demographic questions during app sign-up – who the target reader is (self, child, student, other) and whether the reader is a native speaker of English. Non-native speakers using the app themselves is the largest group, followed by native-speaking children. Study participants in schools and summer camps were predominantly upper elementary students (grades 3-5) in the North-East of the USA at schools and camps catering to majority African American and Hispanic students. Different books were added to the library at different times and received more or fewer readings, depending on study designs and reader interest.

We start with a subset of the data with reasonably complete readings, that is, recordings where at least 70% of the words of the passage were found in the automated transcription (reading accuracy $\geq$ 70%). The 70% cutoff helps filter out data that is unlikely to be useful for studying reading errors, for two reasons: (a) Low accuracies often correspond to cases where large stretches of the passage are left unread (skipped) or to very noisy recordings; feedback in such cases, if any, might have to focus on improving engagement in the reading activity or on improving the quality of recordings, rather than on mispronunciation of specific words. (b) The automated speech analysis is less reliable on low-accuracy recordings (Beigman Klebanov and Loukina, 2021). More information about the system cam be found in Loukina et al. (2019). The system produces fluency estimates that correlate with those obtained using human transcribed data at $r = 0.94$ for recordings above the 70% cutoff (Beigman Klebanov and Loukina, 2021).

The resulting dataset (**ByPassage**) consists of 9,432 recordings by 293 readers of 2,009 unique passages from 24 books. These recordings cover 7,511 word types (unique words) and 136,450 word tokens (all occurrences of the words). Table 1 shows descriptive statistics. Average passage length is 109 words (sd = 37.5) and average accuracy is 91.1% (sd = 8.2). The population distribution of the recordings is: 73.1% school, 9.5% summer camp, and 17.4% users-in-the-wild.

To detect miscues, we use the automated alignment of the recording to the text of the passage generated as part of the accuracy computation (Loukina et al., 2019); we consider as miscues all deletions of words in the passage and all substitutions of words in the passage with other words; insertions of words that were not in the passage were ignored.

4 Patterns of Miscue Occurrence

Reading accuracy, namely, the proportion of words read correctly out of all words in a passage, averaged 91.1%. That is, readings of about 9 in 100 words are miscues; this answers RQ1.

To answer RQ2, we investigate whether miscues tend to cluster together. To determine the proximity of errors to one another, we cluster errors occurring within five tokens of each other with the condition that tokens in a cluster must be part of the same paragraph. Thus, the sequence ECEECCCCECCE, where E stands for error and C for correct, will be considered as one cluster, since there is no stretch of more than four Cs in the sequence. We find that while 32.4% of errors occur singly, most errors are proximal to other errors (see Table 2). On average, clusters have 3.6 errors and span 4 tokens; see Table 3. Thus, errors tend to occur immediately next to each other; patterns like ECCCECCCE are uncommon. Since there are, on average, 9.8 errors per passage and these tend to occur in clusters of 3.6, an average passage would contain 2 or 3 error clusters. The following examples, from *Pinocchio* and *Hansel and Gretel*, respectively, show typical occurrences, with cluster boundaries enclosed in brackets and miscued words denoted in bold:

1. And growing angrier each moment, they went from words to blows, and **[finally began** to **scratch]** and bite and slap each other.
2. The man's **[heart]** smote him heavily, and he thought: "Surely it would be better to share the last **[bite with one's]** children!"

We observe that 5.2% of the errors occur in clusters of 11 errors or more (see Table 2), with the largest cluster consisting of 57 errors. Inspection of the largest cluster, which occurs after about four minutes of reading, reveals that the reader did not read aloud the final paragraph of a long passage.

5 Extended Reading

The app contains a mix of short and long stories, including novels, such as *The Adventures of Pinocchio* and *The Wonderful Wizard of Oz*, each with about 40K word tokens. A novel is different from a sequence of short stories that amount to a similar overall word count in that there tends to be continuity of characters, relationships, and settings throughout the story, with the corresponding repetition of key vocabulary. For example, the word *marionette*, a generally infrequent word, repeats 185 times in *Pinocchio*. Such frequency of occurrence, sometimes in narrator turns and sometimes in reader turns, would provide a lot of opportunities for readers to hear the model performance of the word as well as to practice reading it themselves. The interleaved reading activity itself thus constitutes a kind of feedback to the reader, albeit not immediate and indirect: Frequently occurring words may be self-corrected in subsequent encounters, perhaps making immediate corrective feedback to the reader unnecessarily intrusive.

For our next analysis, we use readings from readers who completed *Pinocchio*, the most read book in the app. For every word type in the book, we collect all its readings from those readers who misread it at least once; these are readers who have correction potential since they made a mistake on the word. Words with fewer than five such readers are discarded. The dataset **Pinocchio** has 19,763 readings of 631 word types read by 47 readers.

Each point in the plot in Figure 1 corresponds to a word type; the size of the dot corresponds to the total number of readers with correction potential. On the x-axis is the $\log_2$ total number of occurrences of the word in the book. On the y-axis is the proportion of readers with correction potential who had at least one correct reading of that word. We start with x = 1, since for x = 0 (one occurrence in the book) it is always the case that y = 0.

Statistic	% Correct	#Words Read Correctly	#Words Read Incorrectly	#Words in the Passage
Mean (SD)	91.1 (8.2)	99.3 (35.5)	9.8 (10.0)	109.1 (37.5)
Mode	100	90	0	99
[Min, Max]	[70.1, 100]	[4, 443]	[0, 74]	[5, 444]
[25%, 50%, 75%]	[85.9, 93.5, 98.0]	[79, 94, 114]	[2, 7, 15]	[90, 101, 124]

Table 1: Descriptive statistics of the reading passages (ByPassage dataset), N = 9,432.

#Errors	Freq.	%	Cumulative %	#Errors	Freq.	%	Cumulative %
1	8,284	32.4	32.4	7	889	3.5	88.8
2	5,019	19.6	52.0	8	651	2.5	91.3
3	3,493	13.7	65.7	9	506	2.0	93.3
4	2,302	9.0	74.7	10	388	1.5	94.8
5	1,559	6.1	80.7	≥11	1,329	5.2	100
6	1,162	4.5	85.3				

Table 2: Distribution of error clusters (N = 25,582) by number of errors in the cluster.

The Figure suggests that generally the more occurrences in the book, the higher the chances of readers figuring out the correct reading even without explicit corrective feedback. We observe that the area to the right of x = 4.32 (20 occurrences or more) and under y = 0.9 (<90% of readers with correction potential with at least one correct reading) is empty, with the exception of the word *would*. As a rough estimate, it seems that about 20 occurrences suffice for the word to largely stop being a problem. We checked this threshold on *The Wizard of Oz* data extracted similarly to the *Pinocchio* data (11,224 readings of 480 word types read by 36 readers) and found it violated by only two words.

These observations suggest that we may want to concentrate the explicit corrective feedback on words that do not occur frequently enough in the book to make self-correction through repeated exposure a near certainty. This would mean that the actual proportion of miscues that are candidates for explicit feedback to the reader may be lower than the 8.9% overall estimate. Removing words with at least 20 occurrences in a story from the list of candidates for explicit feedback for that story, we observe that the proportion of feedback-eligible miscues goes down from 8.9% of all word tokens to 3.3%, for the ByPassage dataset. For an average passage of 109 words, this would correspond to about 3.5 miscues eligible for correction per passage, on average, instead of 9.8. This reduction in the number of miscues eligible for correction is an affordance of the extended reading context; this finding, therefore, answers RQ3.

6 Reliability of miscue detection

Before designing feedback to readers or teachers based on automatically detected miscues, we estimate how reliably the system points out miscues (RQ4). In particular, our focal measure is precision of miscue detection – if a system declares an error, which would presumably trigger feedback, how often is there indeed an error?

We considered words with 50% or lower %Correct, reasoning that these would be likely loci for error flagging. There were 87 such words that were read by at least 10 readers each. We excluded 12 non-dictionary words that may not have a standard pronunciation.[3] Table 4 shows the statistics of the **ByMiscue** sample. These words are generally infrequent, occurring no more than 6 times in the corpus of 24 books. Table 5 lists the words, the number of readings and readers per word, and the titles of the books that included the words.

For every one of the 75 word types, we randomly sampled 3 readings where the machine classified the reading as ‘correct’ and 3 readings classified as ‘incorrect’. In cases with fewer than 3 predicted ‘correct’s, we used all the instances the machine deemed ‘correct’ (2 or 1). There were 446 cases in total for the 75 words, of which 221 had the machine’s prediction of ‘correct’ and 225 ‘incorrect’.

A trained linguist with experience in analysis

[3] The system used human-provided phonetic transcriptions for these words as ‘correct’ pronunciations during the recognition step, but deviation from that may not be clear-cut cases of miscues. These were the excluded words: ’I, ’this, E, h’m, pep-pe, tchee, zik, ziz-zy, zum, zuz-zy, pi-pi-pi, sha’n’t.

Statistic	#Errors per cluster	Cluster span (#words)	#Clusters per passage
Mean (SD)	3.6 (3.6)	4.0 (4.2)	2.7 (2.1)
Mode	1	1	2
[Min, Max]	[1, 57]	[1, 58]	[0, 14]
[25%, 50%, 75%]	[1, 2, 5]	[1, 3, 5]	[1, 2, 4]

Table 3: Descriptive statistics of # errors and # consecutive word tokens in error clusters (cluster spans).

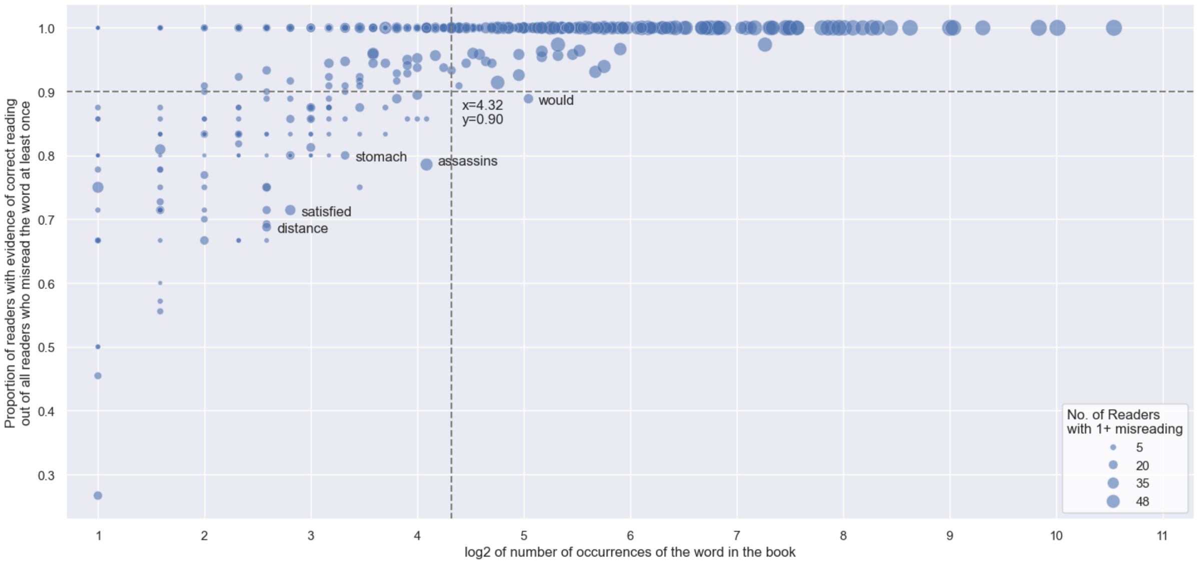


Figure 1: Plot of the relationship between the frequency of a word's occurrence in *Pinocchio* and the proportion of readers who provided a correct reading for the word out of all readers who misread the word at least once. n = 631.

Statistic	Readers	Readings	Tokens
Mean	19.00	21.01	1.95
Median	15	16	1
Mode	10	10	1
SD	10.52	13.40	1.27
Min	10	10	1
Max	55	66	6

Table 4: Descriptive statistics for the ByMiscue sample that covers 75 of the most misread word types.

of oral data (one of the authors of the paper) has listened to the 446 recordings of passages containing the target words, and marked the readings of the target word as 'correct' or 'incorrect'. Table 6 shows the human-machine confusion matrix. The human rater could not make a judgment for 10 instances; these all show as disagreements, equally split between the off-diagonal cells. For 'incorrect' classifications, machine precision was 0.66, recall was 0.65, and the F1 score was 0.66. Thus, about 1 in 3 'incorrect' classifications are false positives – predicting error where there was none.

While performing the annotation, we observed that even when the final execution of a word was correct, there were often indicators that the reader was having some difficulty, such as pausing right before or right after the word, making one or more mistakes leading to the word, or repeating part of the word (e.g., *a fresh convul-convulsion seized her*). The reader's difficulties may manifest in the acoustic signal and, in turn, make it more difficult for the machine to tell whether the reading was correct or incorrect.

We therefore considered a different construct for analysis – that of 'evidence of difficulty' vs. 'no evidence of difficulty' – for the human classification. All instances marked by the human as 'incorrect' in the previous round were labeled as 'evidence of difficulty' by default, whereas the 'correct' cases were further classified into cases with or without evidence of difficulty. Comparing human classification of 'evidence of difficulty' / 'no evidence of difficulty' to the machine's 'incorrect' / 'correct' classification (see Table 7), we found that in 80% of the cases where the machine declared an 'incor-

Word	% C.	R-ngs (R-rs)	Bks
inseparable	10	10(10)	G
scuttling	10	30(30)	P
aristocratic	20	14(14)	B
melodious	20	10(10)	G
zest	20	10(10)	B
pheasants	21	57(39)	P
intuitions	21	14(13)	B
caressed	23	56(55)	P
convulsion	25	12(12)	B
personified	27	11(11)	G
imperturbably	29	14(14)	B
whitened	30	10(10)	G
Pulcinella	31	39(36)	P
indigestion	32	66(53)	P
gold-piece	33	46(46)	P
certainty	33	12(12)	B
Turkish	33	12(12)	B
ventriloquist	33	15(15)	O
astonishment	35	20(20)	PBGR
deductions	36	14(13)	B
sewn	36	14(14)	O
distinguishing	36	11(11)	G
immoderately	36	11(11)	G
sagacity	36	11(11)	G
bedgraggled	37	30(30)	P
bewilderment	37	19(17)	BO
forbearance	40	10(10)	G
persecutors	40	25(25)	P
saucily	40	15(15)	O
magicians	40	10(10)	O
disconsolate	40	10(10)	G
studded	40	15(15)	O
assistance	41	17(17)	BG
caress	41	22(22)	P
carabeneers	41	22(22)	P
amusing	41	22(22)	P
spit	41	64(33)	P
convulsed	42	12(12)	B
excursion	42	12(12)	B
pauper	43	14(14)	B
maliciously	43	21(21)	H
writhed	43	21(13)	GB
slats	43	14(14)	O
partridges	44	27(27)	P
deceived	44	16(16)	O
satin	44	16(15)	NO
perspiration	44	25(24)	P
exquisite	45	31(21)	EG
singed	46	11(11)	O
digested	46	26(14)	P
mastiffs	46	22(22)	P
perpendicular	46	11(11)	G
pursuers	47	19(17)	P
courteously	47	32(30)	PE
sensibly	47	36(36)	P
ferocious	47	15(15)	B
Hippodamia	47	19(14)	G
mysterious	48	21(21)	P
jeeringly	48	21(21)	H
tinsmiths	48	29(14)	O
exhausted	48	31(25)	PAO
spectacle	49	63(52)	P
fancied	49	33(28)	PGR
reproached	50	22(22)	H
perplexity	50	16(16)	GO
brocaded	50	10(10)	O
countless	50	12(12)	O
crocuses	50	14(14)	B
disgustedly	50	22(22)	P
keenly	50	10(10)	P
distinctly	50	10(10)	G
severely	50	12(12)	O
mosquito	50	16(16)	P
stammering	50	24(21)	P
sprightliness	50	10(10)	G

Table 5: 75 most miscued words. %C.: % Correct readings. R-ings(R-rs): #Readings (#Readers). Bks: the source books, from Project Gutenberg: *The Adventures of Pinocchio* by Collodi (P), *the Wonderful Wizard of Oz* by Baum (O), *The Gorgon's Head* by Hawthorne (G), *The Adventure of the Speckled Band* by Conan Doyle (B), *Hansel & Gretel* by Lang (H), *The Necklace* by Maupassant (N), *The Emperor's New Clothes* by Lang (E), *Martin Guerre* by Dumas (A), *Pride & Prejudice* by Austen (R).

Human Machine	Correct	Incorrect
Correct	142	79
Incorrect	77	148

Table 6: Confusion matrix for correct/incorrect human vs machine classification.

Human Machine	No Evidence of Difficulty	Evidence of Difficulty
Correct	106	115
Incorrect	44	181

Table 7: Confusion matrix where machine's correct/incorrect classification is compared the the human's no evidence of difficulty / evidence of difficulty classification.

rect' reading, the human annotator found 'evidence of difficulty' (precision = $\frac{181}{181+44}$ = 0.80); recall was 0.61, and the F1 score was 0.70. Thus, the machine's prediction of 'incorrect' is capturing the human construct of 'evidence of difficulty' with higher precision than the human construct of an 'incorrect' reading.

To confirm the reliability of these findings, a second annotator unrelated to the project with a master's degree in applied linguistics and prior experience annotating speech and oral reading data annotated a reliability sample of 90 randomly selected recordings out of the 446 (about 20%) for (1) correctness of the reading of the target word, and (2) for those items marked as correct, whether there is evidence of difficulty (Appendix A shows the annotation protocol). Cohen's κ between raters for the 3-way classification (incorrect, correct with evidence of difficulty, correct without evidence of difficulty) was 0.604; it was nearly the same (0.601) for a binary classification where 'incorrect' and 'correct with evidence of difficulty' were combined into a single 'evidence of difficulty' class and contrasted with the 'correct with no evidence of difficulty' class.

Using the 90 instances annotated by the second annotator, we also confirmed that the machine's precision was higher in detecting the second rater's 'evidence of difficulty' annotations than the second rater's 'incorrect' annotations (precision of 90% for 'evidence of difficulty' and 84.2% for 'incorrect'). The precision for the first annotator's data for the same subset of 90 instances was 84.2% vs 76.3% for the two constructs, respectively.

To summarize: Our analyses suggest that the machine's detection of a miscue corresponds more precisely to what a human listener would consider as a reading showing evidence of difficulty (80.4% precision) than to what a human listener would designate as a miscue (65.8% precision). This is because readers sometimes ended up reading the word correctly, perhaps after an initial stumble or a partial reading, or recovering from a misreading of a few words just before the current one; the machine often did not recognize these as correct readings.

7 Discussion: Implication for feedback ideation

Our analysis of the automatically transcribed read aloud data from an interleaved book reading app shows a substantial extent of reading difficulty in the readers: About 9% of all word readings in the eligible transcripts show evidence of difficulty based on an automated analysis. The actual extent of difficulty, as detected by a human listener, is likely to be higher, since, while the system shows fairly high precision (0.80) in detecting what a human listener would consider evidence of difficulty, it misses many such cases, since the recall stands at 0.61. Inspecting the patterns in about 9.5K recordings by 293 readers of 2K unique passages (excerpts from novels and short stories), we observed that reading difficulties tend to cluster in 3-4 consecutive words, suggesting that corrective feedback to the reader may need to contain a model performance of whole phrases rather than individual words.

Further, we examined the interleaved extended reading and listening itself as a kind of delayed (not immediate) and indirect feedback to the reader that does not require to break the flow of reading. We estimated that a word that occurs 20 times or more in the book is likely to have sufficient exposure in narrator and reader turns for 90% of the readers who misread it at least once to also produce at least one correct reading. Assuming that there is no urgency that the reader learn a particular word now instead of a few chapters later, we may want to forgo giving the reader direct feedback on misreadings of words that will almost certainly get fixed by the time the reader finishes the story, focusing instead on misreadings of words that do not get repeated very often in the story.

Finally, when designing the feedback based on automatically detected misreadings, it is important to keep in mind that, at least with the speech recognition technology currently implemented in the app and the type of data typical of this use case (no acoustic control of the environment, consumer-level devices and headsets), the detection of miscues is only 66% precise.[4] However, in 80% of the cases where the machine flags a miscue, there is evidence that the reader is having some difficulty – whether or not they produced a correct reading in the end. The ideation and design of feedback will need to reflect this shift in the construct. This finding also suggests that, in terms of learner modeling, automatically detected reading errors provide evidence not only on the knowledge dimension, but also on a behavioral dimension – miscues flagged by the system may provide a first-cut detection of loci where evidence of multiple attempts, self-corrections, pausing to consider the difficult word, and other behaviors related to the trait of perseverance may be found, upon further analysis.

As a first step in exploring feedback to the teacher based on evidence of difficulty, we created class-level heatmaps per paragraph for an ongoing reading of *Pinocchio* in a 4th grade classroom and sent the teacher the heatmaps for the paragraphs that were most difficult for the class, one per chapter. In an interview, the teacher described her use of the heatmap shown below. She told the students she was showing them a challenging passage and explained the darker red as standing for more readers having a difficulty. She told students that some of it was a bit of a tongue-twister for any reader (she said she would have had a hard time herself); she then praised the class for reading much of the passage well and for giving the more challenging part a go. The class also had a brief discussion of what *a gold-piece indigestion* meant. The teacher thus used the feedback not only for providing a correct reading of the miscue cluster “gold-piece indigestion" that occurred in many of her students’ readings, but also for a brief but rich motivational, affective, and comprehension-related activity.

> Pinocchio ate **least** of all. He asked for a bite of bread and a few nuts and then hardly touched them. The poor fellow, with his mind on the Field of Wonders, **was suffering from** a **gold-piece indigestion**.

8 Conclusion

In this paper, we analyzed miscues detected by an automated speech analysis system deployed through a publicly available reading app where readers take turns reading books out loud with a pre-recorded skilled narrator. The impetus for considering miscues came from teachers’ request to provide such data; however, it is not a-priori clear (a) what the extent of the target behavior is; (b) in what patterns it occurs that may suggest certain ways of designing feedback, (c) how the design of the reading activity may impact feedback, and (d) exactly what a flagged miscue is communicating.

Our analysis shows that miscuing, or, rather, readers experiencing some difficulty reading the word, even if they do pronounce it correctly in the end, is extensive – about nine in a hundred words and possibly more, since our automated system does not detect all such cases. Second, problems tend to cluster together, suggesting that corrective feedback may better be presented by modeling the reading of a phrase rather than of individual words. Third, in order to minimize the interference with the flow of reading, one may want to prioritize modeling misread words that do not occur very frequently in the book. We found that words with 20 or more repetition are very likely to be learned through the interleaved reading activity itself, without additional explicit feedback.

Finally, examining the reading instances flagged by the system as miscues, we found that these are not necessarily incorrectly read words but is closer to what a human listener would consider as evidence of some difficulty on the reader’s part, whether or not the word came out correct in the end. This opens up the possibility of considering the automatically detected miscues as a first-cut detection of instances of reader struggles – not only those that manifest as an error but also those that show gearing-up or preparation (pause before), persistence (multiple attempts), or self-correction (successful final readings) – all providing evidence not only on the skill dimension, but also on important learner traits such as perseverance.

From the point of view of feedback development, our analyses suggest that when designing feedback to the reader, it may be incorrect to start

[4] It is interesting that the system nevertheless provides reliable fluency estimates – estimates of words read correctly *per minute*. It may suggest that the impact of crediting or not those words that came out correctly after some struggle is relatively small, considering that the struggle itself has taken time without emission of correct words, which is appropriately captured as detrimental to fluency.

from the vision of “give feedback on every miscue." First, there may be too many of them. Second, it would make sense to fuse feedback on multiple miscues since they tend to cluster together. Third, some of the miscued words have a verifiably high chance of getting fixed during the activity without explicit feedback. Finally, the content of the feedback would not actually be a correction of a miscue, because there may not have been an actual miscue – or misreading – to begin with; it may have been a successful reading following some struggle. This shift in the construct suggests feedback not only, or not necessarily, along the dimension of reading skill, but also learner traits such as perseverance. Our first trial with a teacher shows promise in that the teacher was able to use the evidence of difficulty feedback not only for a corrective purpose, but also for a motivational and affective one.

More broadly, our case study shows how a detailed examination of existing data from a new angle may provide new insights into the performance of the system and support ideation of a new use of the system to the benefit of the stakesholders.

9 Limitations

In this study, data was not separated by characteristics of readers that might impact the kind of mistakes they are making during reading. For example, we did not consider the possible effects of age, linguistic background, or learning disabilities, since we know relatively little about users-in-the-wild and about readers in informal contexts, beyond the general description provided earlier. In addition, it is possible that the automated system performs more accurately on data from certain kinds of readers than from others. For example, recordings from soft-spoken readers or readers with speech disorders may be more difficult to analyze accurately. Different kinds of performances may also be easier or harder to handle; we have anecdotal evidence that particularly creative performances – such as a reader singing the passage – might be difficult for automated analysis.

In the current study, we investigated relatively large-scale patterns in order to identify important considerations for feedback ideation; specific designs will need to be informed by more nuanced analyses of use cases, user populations, and personal reading histories of the users of the application.

10 Ethics statement

Data collections at all the school and summer camp sites were approved by our institution’s IRB. The users-in-the-wild agree to the Terms of Use (`https://relayreader.org/terms`) during sign-up into Relay Reader, including the following statement that appears on the Terms of Use summary page displayed prominently during sign-up: “ETS collects voice recordings and other data from users of the App. The recordings and usage data are used in an anonymized manner in connection with ETS research," followed by a link where more information about the research can be obtained. If the application is being installed by parents or teachers for their children and students, respectively, the following statement (that also appears in the Terms of Use summary) additionally applies: “If I am downloading this App for use by my child or student, I have the authority to permit ETS to collect the recordings and usage data as described in the Terms of Use." Our organization’s Privacy Policy is linked from relayreader.org and is available here: `https://www.ets.org//legal/privacy.html`.

The data is oral reading data of stories in the Relay Reader app and process data from the app. As such, it is not expected to contain content such as the reader’s name, thoughts or opinions, and, indeed, this has not been observed in the data inspected in detail (ByMiscue sample). We have not taken additional steps to check whether the data that was collected contains any information that names or uniquely identifies individual people or offensive content. The data collected by the app is securely stored and managed in accordance with our organization’s Privacy Policy.

All the stories and narrations used in the App are either in the public domain (in which case the texts are sourced from Project Gutenberg and the narrations are sourced from LibriVox.org, a collection of volunteer public domain recordings of public domain books), or licensed from the copyright holders.

Acknowledgements

We would like to thank Dr. Burcu Arslan for her insightful comments on an early draft of this paper.

References

Marilyn Jager Adams. 2013. The promise of automatic speech recognition for fostering literacy growth in children and adults. In *International handbook of literacy and technology*, pages 109–128. Routledge.

Scott Ardoin, Katherine Binder, Tori Foster, and Andrea Zawoyski. 2016. Repeated versus wide reading: A randomized control design study examining the impact of fluency interventions on underlying reading behavior. *Journal of School Psychology*, 59:13–38.

Yu Bai, Ferdy Hubers, Catia Cucchiarini, and Helmer Strik. 2020. ASR-based evaluation and feedback for individualized reading practice. In *Proceedings of INTERSPEECH*, pages 3870–3874.

Beata Beigman Klebanov and Anastassia Loukina. 2021. Exploiting structured error to improve automated scoring of oral reading fluency. In *Proceedings of the International Conference on Artificial Intelligence in Education (AIED)*, pages 76–81.

Ronald Cole, Barbara Wise, and Sarel van Vuuren. 2007. How Marni teaches children to read. *Educational Technology*, 47(1):14–18.

Lucile Gelin, Morgane Daniel, Thomas Pellegrini, and Julien Pinquier. 2023. Comparing phoneme recognition systems on the detection and diagnosis of reading mistakes for young children's oral reading evaluation. In *Proceedings of Speech and Language Technologies in Education (SLaTE)*, pages 6–10.

Lucile Gelin, Morgane Daniel, Julien Pinquier, and Thomas Pellegrini. 2021. End-to-end acoustic modelling for phone recognition of young readers. *Speech Communication*, 134:71–84.

Andreas Hagen, Bryan Pellom, and Ronald Cole. 2007. Highly accurate children's speech recognition for interactive reading tutors using subword units. *Speech Communication*, 49(12):861–873.

Corentin Hembise, Lucile Gelin, and Morgane Daniel. 2021. Lalilo: A reading assistant for children featuring speech recognition-based reading mistake detection. In *Proceedings of INTERSPEECH, Show & Tell contribution*.

Alida Hudson, Poh Koh, Karol Moore, and Emily Binks-Cantrell. 2020. Fluency interventions for elementary students with reading difficulties: A synthesis of research from 2000–2019. *Education Sciences*, 10(3):52.

Jeffrey Huffman. 2014. Reading rate gains during a one-semester extensive reading course. *Reading in a Foreign Language*, 26(2):17–33.

Priya Kannan, Beata Beigman Klebanov, Shiyi Shao, Colleen Appel, and Rodolfo Long. 2019. Evaluating teachers' needs for ongoing feedback from a technology-based book reading intervention. In *Paper presented at the Annual Meeting of the National Council on Measurement in Education (NCME)*, Toronto, ON, Canada.

Anastassia Loukina, Beata Beigman Klebanov, Patrick L Lange, Yao Qian, Binod Gyawali, Nitin Madnani, Abhinav Misra, Klaus Zechner, Zuowei Wang, and John Sabatini. 2019. Automated estimation of oral reading fluency during summer camp e-book reading with My Turn To Read. In *Proceedings of INTERSPEECH*, pages 21–25.

Nitin Madnani, Beata Beigman Klebanov, Anastassia Loukina, Binod Gyawali, Patrick L 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.

Jack Mostow and Gregory Aist. 1999. Giving help and praise in a reading tutor with imperfect listening — because automated speech recognition means never being able to say you're certain. *CALICO Journal*, 16(3):407–424.

John Pikulski and David Chard. 2005. Fluency: Bridge between decoding and reading comprehension. *The Reading Teacher*, 58(6):510–519.

Sherry Ruan, Angelica Willis, Qianyao 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 Learning@Scale*, pages 1–4.

John Sabatini, Zuowei Wang, and Tenaha O'Reilly. 2019. Relating reading comprehension to oral reading performance in the NAEP fourth-grade special study of oral reading. *Reading Research Quarterly*, 54(2):253–271.

Namhee Suk. 2017. The effects of extensive reading on reading comprehension, reading rate, and vocabulary acquisition. *Reading Research Quarterly*, 52(1):73–89.

Jade Wexler, Sharon Vaughn, Meaghan Edmonds, and Colleen Klein Reutebuch. 2008. A synthesis of fluency interventions for secondary struggling readers. *Reading and Writing*, 21:317–347.

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 *Proceedings of INTERSPEECH*, pages 1–5.

A Annotation protocol

Use the following coding scheme to classify how each word was read:

Correct A word was read correctly, without difficulty.

Correct with difficulty A word was read correctly, even if initially read incorrectly, or with other signs of difficulty.

Incorrect A word was read incorrectly, even if initially read correctly.

A.1 Features of incorrect reading

A word should be coded as **incorrect** if it has any of the following qualities.

Mispronunciation Part of the word is not pronounced as it should be expected. In the case of proper nouns, any reasonable phonetic pronunciation of the word is acceptable. Mispronunciations can include: (a) Pronouncing the wrong segment, e.g., saying SUN as SOON, saying NATION as NATE-EE-ON; (b) Inserting an extra syllable, e.g., saying NATION as NA-SHE-ON; (c) Omitting part of a word such as a segment, syllable, or suffix, e.g., saying WISH instead of WISHES, saying DESCRIBED as DESCRIDE, saying AMBIGUOUSLY as AMBIGUSELY; (d) Reversing the order of segments, e.g., saying ONIMOUS instead of OMINOUS; (e) Using the wrong lexical stress pattern, e.g., saying JAPANESE as JA**PAN**ESE.

Replacement The reader says a different word instead of the target, e.g., IMPERIAL instead of EMPIRICAL, AUTOMOBILE instead of AUTOMATIC.

Not blending The reader sounds out the individual segments in a word instead of blending them together.

Intra-word pausing The reader pauses for an extended period of time mid-word, especially at a point that is not near an inflectional suffix or in a way that reduces intelligibility. e.g., ELE ... PHANT, TER ... MINATE.

Subvocalization The reader makes noises that resemble the word, such as by pronouncing a few segments while grunting or mumbling the rest.

A.2 Features of correct reading with difficulty

If a word has none of the features of incorrect reading, it should be coded as **correct with difficulty** if any of the following occur in or around the word.

Pausing The reader **unnaturally** pauses before or after the word at a point where the pausing is expected to be caused by difficulty with the word, such as: (a) Immediately before or after the word; (b) At a phrasal or clausal boundary before the word, in a manner where it does not seem that the difficulty is associated with another word.

Errors near the word The reader reads one or more word incorrectly before or after the word. This may be in an adjacent word or up to 4 words before or after the word if the errors do not seem to be caused by another difficult word nearby. This classification would occur, for example, if "immoderately" were targeted for analysis, and the reader omitted "so" but read "immoderately" correctly when reading "In short, she is so **immoderately** wise people call her wisdom personified...".

Repetition The reader says the word multiple times and says the final attempt correctly. The initial attempts may be either correct or incorrect.

False start The reader says part of the word, stops, and says the word again from the beginning correctly, e.g., saying CONCERT as CON-CONCERT.

Intonation The reader uses rising intonation on the word, as if asking a question, in a manner that expresses uncertainty about correctness of the reading.

Mumbling The reader is mumbling through the part where the word occurs, perhaps subvocalizing several words, but reads the target word correctly.

Findings from the First Shared Task on Automated Prediction of Difficulty and Response Time for Multiple-Choice Questions

Victoria Yaneva[1], Kai North[2], Peter Baldwin[1], Le An Ha[3], Saed Rezayi[1], Yiyun Zhou[1], Sagnik Ray Choudhury[1], Polina Harik[1], and Brian Clauser[1]

[1]National Board of Medical Examiners, Philadelphia, USA
{vyaneva, pbaldwin, srezayidemne, yyzhou, sraychoudhury, pharik, bclauser}@nbme.org
[2]George Mason University, USA
knorth8@gmu.edu
[3]Ho Chi Minh City University of Foreign Languages, Vietnam
anhl@huflit.edu.vn

Abstract

This paper reports findings from the First Shared Task on Automated Prediction of Difficulty and Response Time for Multiple-Choice Questions. The task was organized as part of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA'24), held in conjunction with NAACL 2024, and called upon the research community to contribute solutions to the problem of modeling difficulty and response time for clinical multiple-choice questions (MCQs). A set of 667 previously used and now retired MCQs from the United States Medical Licensing Examination (USMLE®) and their corresponding difficulties and mean response times were made available for experimentation. A total of 17 teams submitted solutions and 12 teams submitted system report papers describing their approaches. This paper summarizes the findings from the shared task and analyzes the main approaches proposed by the participants.

1 Introduction

For standardized exams to be fair and defensible, test items must meet certain criteria. One important criterion for many exams is that the questions cover a wide range of difficulty levels to allow information about a wide range of examinee proficiencies to be collected effectively. Additionally, it is often essential to allocate an appropriate amount of time for each question: too little time can make the exam speeded, while too much can make it inefficient. Often, item difficulty and response time data are collected via a process called *pretesting*, wherein new items appear on live exams alongside scored items. While robust, the need for a statistically sufficient sample of examinees to complete these items restricts the number of items that can be pretested, potentially leading to overexposure and jeopardizing item security (Settles et al., 2020).

The problem of estimating item characteristics with little to no response data is a decades-old research topic. Early studies used what is sometimes referred to as auxiliary or collateral information—including various properties of an item's text—to improve parameter estimation within a Bayesian framework (Mislevy, 1988; Stowe, 2002; Swaminathan et al., 2003). Recent advances in NLP have led to a renewed interest in predicting item characteristics based on item text. As with the earlier research, it is hoped that such predictions may be used to "jump-start" parameter estimation (McCarthy et al., 2021) allowing items to be exposed to fewer test-takers, or improve fairness by making the time intensiveness of test forms that include pretest items less variable (Baldwin et al., 2020).

While there is evidence that NLP techniques may offer a potential solution (see Section 2), the absence of publicly available datasets has resulted in fragmented efforts to advance the state-of-the-art in item parameter prediction, impeding meaningful comparisons between different approaches, exacerbating issues of reproducibility, and stifling collaboration. Furthermore, as outlined in Section 2, the existing literature has concentrated on difficulty prediction, neglecting other crucial item parameters such as response time, which also have important implications for exam fairness and validity.

To address these shortcomings and advance this area of research, we organized the First Shared Task on Automated Prediction of Difficulty and Response Time for Multiple Choice Questions[1]. The shared task was organized as part of the 19th

[1] https://sig-edu.org/sharedtask/2024

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 470–482
June 20, 2024 ©2024 Association for Computational Linguistics

Workshop on Innovative Use of NLP for Building Educational Applications (BEA'24), collocated with NAACL 2024, and took place between January 15 and March 10, 2024. An ideal shared dataset for this task would encompass test items along with their corresponding difficulties and response times based on responses collected from a sufficiently large and diverse examinee sample under standardized test conditions. To this end, 667 retired clinical multiple-choice questions (MCQs) from a high-stakes medical exam[2] were released for the exploration of two topics: predicting item difficulty (Track 1) and predicting item response time (Track 2). Overall, 48 teams enrolled as participants, of which 17 submitted solutions and 12 submitted system review papers describing their approaches. This paper summarizes the organization and main findings from the competition. The data are available upon request at https://www.nbme.org/services/data-sharing.

2 Related Work

This section summarizes the main approaches used in item difficulty and response time prediction research, with special emphasis on clinical MCQs, the domain of the shared task. For a systematic review of the literature, we refer the reader to AlKhuzaey et al. (2023).

2.1 Predicting Item Difficulty

Most of the early research on modeling item difficulty was in the domain of language learning and used predictors such as lexical, syntactic, statistical, and readability features. Freedle and Kostin (1993) and Perkins et al. (1995) used a mix of lexical and syntactic features, such as vocabulary, sentence and paragraph length, number of negations and referentials, and lexical overlap between text and options to determine the difficulty of MCQs from English foreign language exams and reading comprehension tests, respectively. These features were later expanded to cohesion, discourse, and psycholinguistic features among others (Beinborn et al., 2014, 2015; Loukina et al., 2016).

Outside the domain of language learning, these features showed comparatively weaker predictive power. El Masri et al. (2017) found that linguistic features were not good predictors for item difficulty in middle-school science items, "likely due to the extent to which computational linguistic facilities are less effective with very short textual materials". Likewise, Susanti et al. (2017) and Benedetto et al. (2020) found that readability metrics were relatively poor predictors of item difficulty for computer science and English vocabulary MCQs, respectively.

Consistent with other NLP use cases, more recent studies on item parameter prediction utilize neural approaches. Huang et al. (2017) used embeddings and an attention-based convolutional neural network to predict the difficulty of reading items. Hsu et al. (2018) converted items into word-embeddings, calculated the cosine similarities between stem, answer, and distractors, and used them to train a support vector machine (SVM) to predict item difficulty of MCQs from the domain of social studies. Zhou and Tao (2020)'s fine-tuned BERT model (Devlin et al., 2018) achieved a higher F1-score for predicting item difficulty of open-ended programming-related questions compared to a Bidirectional Long Short-Term Memory (BiLSTM) model. Benedetto et al. (2021) trained a series of BERT and DistilBERT models with several pre-training steps, including the use of masked-language modeling. BERT achieved the highest performance for predicting item difficulty of math and computer science open-ended questions and MCQs, having surpassed all other models—including several word-embedding approaches. Other notable studies in this area include Loginova et al. (2021) and He et al. (2021).

Item difficulty prediction has also been applied in efforts to automatically generate items at desired levels of difficulty (e.g., Gao et al. (2018), Bi et al. (2021)). Some of these approaches assess the semantic similarity between a question and its associated answer choices (Alsubait et al., 2013; Kurdi et al., 2016), while others focus on items that assess an examinee's ability to distinguish between words and pseudo-words, and thus utilize word and sub-word level predictors (Settles et al., 2020).

2.2 Predicting Item Response Time

The prediction of response time is a less-researched area, further motivating its inclusion within this shared task. Early studies included features such as the sequential position of the item within an exam (Parshall et al., 1994), the inclusion of visual aids (Smith, 2000; Swanson et al., 2001), and word-count (Halkitis et al., 1996; Smith, 2000).

[2] The United States Clinical Licensing Examination (USMLE®)

A 65-year-old woman comes to the physician for a follow-up examination after blood pressure measurements were 175/105 mm Hg and 185/110 mm Hg 1 and 3 weeks ago, respectively. She has well-controlled type 2 diabetes mellitus. Her blood pressure now is 175/110 mm Hg. Physical examination shows no other abnormalities. Antihypertensive therapy is started, but her blood pressure remains elevated at her next visit 3 weeks later. Laboratory studies show increased plasma renin activity; the erythrocyte sedimentation rate and serum electrolytes are within the reference ranges. Angiography shows a high-grade stenosis of the proximal right renal artery; the left renal artery appears normal.
Which of the following is the most likely diagnosis?
(A) Atherosclerosis
(B) Congenital renal artery hypoplasia
(C) Fibromuscular dysplasia
(D) Takayasu arteritis
(E) Temporal arteritis

Table 1: An example of a practice item from the USMLE Step 1 Sample Test Questions (`usmle.org`). © 2024 National Board of Medical Examiners and the Federation of State Medical Boards, used with permission.

Schneiderand et al. (2023) is one of the few studies that used text-based features to predict student response time for items on multiple topics, ranging from everyday life to personality and politics. They trained models such as stochastic gradient boosting (SGB), SVM, and random forests (RF) on 51 features including question length, lexical diversity, and readability features, such as number of complex words, with SGB achieving best performance.

2.3 Focus on Clinical MCQs

The studies most relevant to this shared task are the ones focused on predicting characteristics of clinical MCQs from the USMLE exam. These include Ha et al. (2019), who used a 113 linguistic features and different embedding types to predict the difficulty (proportion correct responses) of 12,038 items. This study indicated that predicting item difficulty for this domain is a challenging task, with Root Mean Squared Error (RMSE) of .225 for the best result compared to a dummy regressor baseline of .237. Baldwin et al. (2020) built upon this study by applying the same predictors to the problem of modeling response time, and showed that exam fairness can be improved through meaningful reductions in the variability of time intensiveness across test forms when predicted response times for pretest items are taken into accounted during form assembly. Xue et al. (2020) applied transfer learning to the prediction of item parameters and showed that the prediction of difficulty can be improved by incorporating response time during training, but not vice-versa. Yaneva et al. (2020) aimed to automatically identify items that meet statistical criteria for live use in terms of both difficulty and discrimination[3]. Yaneva et al. (2021) examined the relationship between the linguistic characteristics of a test item and the complexity of the response process required to answer it correctly, defined as the interaction between difficulty and response time. The methods used in these studies are summarized in Yaneva et al. (2023), which was written for educational measurement professionals and provides an overview of the applications of NLP methods to this task.

3 Shared Task Description

The data for the shared task comprises 667 previously used and now retired MCQs from Steps 1, 2 CK, and 3 of the United States Medical Licensing Examination (USMLE®). USMLE is a sequence of examinations (called *Steps*), developed by the National Board of Medical Examiners (NBME®) and Federation of State Medical Boards (FSMB), that is used to support medical licensure decisions in the United States. Each step includes 7 to 12 blocks of MCQ items (a block ranges between 45 and 60 minutes), and each item is answered by approximately 300+ examinees. Item characteristics used in this shared task were based on examinees who were medical students from accredited[4] US and Canadian medical schools taking the exam for the first time.

An example practice item from the dataset is given in Table 1. The part describing the case is referred to as *stem*, the correct answer is referred to as *key*, and the incorrect answer options are known as *distractors*. All items test medical knowledge

[3] Item discrimination is a measure of the extent to which an item differentiates between students of different proficiency.

[4] Accredited by the Liaison Committee on Medical Education (LCME).

and were written by experienced subject-matter experts following a set of guidelines. These guidelines stipulate adherence to a standard structure, as well as the avoidance of extraneous material not needed to answer the item, information misleading the test-taker, or correct answers that are longer or more specific than the other options.

Each item is tagged with metadata indicating whether or not it contains an image, the Step exam it was presented on, as well as Difficulty and Response Time data, as shown in the structure below:

- *ItemNum* denotes the consecutive number of the item in the dataset (e.g., 1,2,3,4,5, etc).
- *ItemStem_Text*: the text of the item stem (the part of the item describing the clinical case).
- *Answer_A*: the text for response option A
- *Answer_B*: the text for response option B
- *(. . .)*
- *Answer_J*: the text for response option J. For items with fewer than J response options, the remaining columns are left blank. For example, if an item contains response options A to E, the fields for columns F to J are left blank for that item.
- *Answer_Key*: the letter of the correct answer.
- *Answer_Text*: the text of the correct answer.
- *ItemType*: whether the item contained an image (e.g., an x-ray image, picture of a skin lesion, etc.) or not. The value "Text" denotes text-only items and the value "PIX" denotes items that contain an image. Note that the images are not part of the dataset.
- *EXAM*: The USMLE Step (1, 2 or 3) the item was presented on. For more information on the Steps of the USMLE see https://www.usmle.org/step-exams.
- *Difficulty*: The (linearly-transformed) proportion of correct responses across all examinees who attempted a given item during a live exam. After the transformation, higher values indicated more difficult items.
- *Response_Time*: arithmetic mean response time, measured in seconds, across all examinees who attempted a given item on a live exam. This includes all time spent on the item from the moment it is presented on the screen until the examinee moves to the next item, as well as any time spent revisiting the item.

The task was divided in two tracks as follows:

- Track 1: Given the item text and metadata, predict the item difficulty variable.
- Track 2: Given the item text and metadata, predict the time intensity variable.

Out of the full sample, 466 items were made available as a labeled training set and the other 201 items were retained as an evaluation set. Training data outside of the specified training set were allowed, provided these data were publicly available and their license allows use for research purposes. Use of one target variable in the prediction of another was *not* permitted, since in most cases, predicting these variables will be most beneficial prior to the collection of response data—at which time neither the difficulty nor the time intensity parameters can be estimated.

Submissions were requested as separate .csv files for each track. Each file had to contain the item number (Item_Num) and corresponding predicted value for each item. Teams were allowed to submit up to three attempts per track, differentiated by adding run1, run2, or run3 to the name of their uploaded .csv file; however, such teams were required to explain how each attempt differed within their system report paper—i.e., changes in methodology, parameters, models used, prediction strategy, etc.

In both tracks, the evaluation was based on RMSE, and teams that achieved the lowest RMSE value were considered winners. There were two separate leaderboards for Track 1 and Track 2. In both, submissions were ranked according to the RMSE metric from Python's scikit-learn library (Pedregosa et al., 2011).

4 Results

A total of 17 teams submitted up to 3 solutions for item difficulty prediction and 15 teams submitted up to 3 solutions for response time prediction. Table 2 presents ranked results for the top 15 solutions in both tracks. The full leaderboard is available at https://sig-edu.org/sharedtask/2024#results.

In Track 1, Predicting Item Difficulty, there are minor differences between the RMSE of the top 15 solutions; however, even the best solution outperformed the baseline by only a small margin (#1, EduTec = 0.299, #16, DummyRegressor = 0.311). These results are consistent with the prior literature

Difficulty				Response Time			
Rank	Team Name	Run	RMSE	Rank	Team Name	Run	RMSE
1	EduTec	electra	0.299	1	UNED	run2	23.927
2	UPN-ICC	run1	0.303	2	ITEC	Lasso	24.116
3	EduTec	roberta	0.304	3	UNED	run1	24.777
4	ITEC	RandomForest	0.305	4	UNED	run3	25.365
5	BC	ENSEMBLE	0.305	5	EduTec	roberta	25.64
6	Scalar	Predictions	0.305	6	EduTec	electra	25.875
7	BC	FEAT	0.305	7	UnibucLLM	run3	26.073
8	BC	ROBERTA	0.306	8	ED	run1	26.57
9	UnibucLLM	run1	0.308	9	Rishikesh	1	26.651
10	EDU	Run3	0.308	10	UnibucLLM	run2	26.768
11	EDU	Run1	0.308	11	UnibucLLM	run1	26.846
12	ITEC	Ensemble	0.308	12	SCaLARlab	run3	26.945
13	UNED	run3	0.308	13	Scalar	predictions	26.982
14	Rishikesh	1	0.31	14	EduTec	deberta	27.302
15	Iran-Canada	run2	0.311	15	EDU	Run1	27.474
16	Dummy Regressor Baseline		0.311	25	Dummy Regressor Baseline		31.68

Table 2: Top 15 leaderboard results for Track 1: Difficulty and Track 2: Response Time

on clinical MCQs presented in Section 2.3, underscoring the challenging nature of the task. In Track 2, Response Time, the solutions are relatively more successful compared to the DummyRegressor baseline (#25 DummyRegressor, RMSE = 31.68), with the #1 solution obtaining RMSE of 23.927.

Of the 17 teams who submitted solutions, 12 submitted system report papers, which are summarized below (10 papers for both Track 1 and Track 2, and 2 papers only for Track 1).

5 Main Approaches

The solutions submitted by the participants encompassed several approaches that had not been previously applied to the problem of modeling item characteristics. Some of these were comparatively simpler models that performed unexpectedly well, such as the case of the submission that ranked #1 in predicting response time (Rodrigo et al., 2024). In the case of modeling item difficulty, several approaches used classical methods such as linguistic features combined with embeddings but expanded the set of features to include novel predictors. These traditional solutions were not as successful for item difficulty prediction, which favored more novel approaches. These novel approaches can be broadly categorized as transformer model modifications, question answering using LLMs, and data augmentation techniques. These categories are not necessarily mutually exclusive (e.g., some approaches use both data augmentation and linguistic features); however, we found this broad classification scheme useful in describing the submitted solutions, as shown below. The main techniques used in the studies are further summarized in Section 5.6.

5.1 Efficient solutions that performed well

Well-performing solutions include the ones proposed by **UNED** (Rodrigo et al., 2024), who focused on feeding combinations of the full item, stem and correct answer, or stem only into a BERT base model (Devlin et al., 2018). The three submissions differed only by these input configurations and were the same for both tracks (with different target variables). There was no special preprocessing and the tokenzier was the one provided by the BERT model. The target variables were both scaled [0-1]. Perhaps somewhat surprisingly given its simplicity, this system ranked #1 for response time prediction (RMSE of 23.927 with text and correct answer as input) and #13 for difficulty prediction (RMSE of 0.308, stem only).

Scalar (DataWizards) concatenated BERT embeddings with TF-IDF encodings for item difficulty prediction and Word2Vec embeddings with TF-IDF encodings for response time prediction. These representations of different item components (e.g., stem only or stem + answer options) were used as predictors in various models, of which RF performed best. This solution ranked #6 for predicting item difficulty (RMSE = 0.305) and #13 for response time (RMSE = 26.982).

These solutions serve as an important benchmark for the added value provided by the linguistic features, question-answering techniques, and model optimization approaches presented next.

5.2 Transformer model modifications

The solution that ranked #1 for predicting difficulty was from the category of novel model optimization techniques. **EduTec** (Gombert et al., 2024) proposed optimizing pre-trained transformer encoder language models using three modifications. The first modification was the use of scalar mixing, which is a procedure that calculates a weighted mean of all hidden layers of the transformer (the weights are fit during training). Scalar mixing is hypothesized to be helpful because, as different layers within transformer models learn representations for different linguistic phenomena, it allows the use of representations from all these different layers (as opposed to the final layer alone), while simultaneously learning their importance for the final output. The second modification was a two-layer setup for the classification heads, where the input from the intermediate layer was run through a *rational activation*: a form of learnable activation function whose shape is optimized during training. This type of activation function was shown to outperform non-learnable activation functions. Third, the authors used multi-task learning to learn shared representations for both difficulty and response time, motivated by the observed correlation between the two variables within the training set. The architecture described so far was evaluated with different transformer encoder models, of which ELECTRA achieved #1 in the shared task leaderboard for difficulty prediction with an RMSE of 0.299 and #6 on the leaderboard for response time prediction (RMSE = 25.875). RoBERTa achieved #5 for response time prediction with an RMSE of 25.64.

5.3 Question answering using LLMs

Two teams used responses from LLMs to extract predictive features or perform data augmentation.

UPN-ICC (Dueñas et al., 2024) investigated the hypothesis that item difficulty depends more on the features of the test-taking population than on the items themselves. They simulated medical students' answers to the MCQs by prompting chatGPT 3.5 in four different settings: i) answering each question and providing a brief justification for the response, ii) providing a yes/no response for each answer option on whether it is the correct answer, iii) randomly removing 20% of the content tokens from the stem to simulate examinees who did not read the item carefully, and iv) all of the above but with a varying temperature parameter[5]. The justification behind iv) is the hypothesis that items that are only answered correctly under a low temperature condition can be considered difficult, while items answered correctly under any temperature can be considered easier. Next, the authors extracted more than 40 features from the generated output of the question-answering experiments. Examples of such features include "A Boolean indicating whether or not the question was answered correctly by the LLM" and "Time in milliseconds reported by the LLM to answer the question" for condition i), "Number of sub-items answered correctly for the item" for condition ii), "Boolean indicating if the LLM answered correctly the question in spite of the stem being mutilated at 20% of its content words (other six features for 30%, 40%, 50%, 60%, 70%, and 80%" for condition iii), and "Number of incorrect answers for the item out of the 11 values of *t* [temperature] used" for condition iv). These features were used as input for a Ridge regression model, which ranked #2 in difficulty prediction (RMSE = 0.303). While the indicator of whether the question was answered correctly emerged as the most significant feature, all four strategies produced meaningful predictors.

UnibucLLM (Rogoz and Ionescu, 2024) hypothesized that the number of LLMs that answer an item correctly can be an indicator of its difficulty. In a zero-shot setup, they obtained responses from three LLMs (Falcon-7B, (Almazrouei et al., 2023), Meditron-7B (Chen et al., 2023), and Mistral-7B (Jiang et al., 2023)). They then created variations of the input that included the item text only or the item text together with the LLM responses. This input was used to finetune a pretrained BERT model and a pretrained GPT-2 model (Radford et al., 2019). The best solution for difficulty prediction was the BERT model finetuned over the item text + the answer text + the LLM-generated answers, which placed #9 with an RMSE of 0.308, showing a positive effect from the LLMs. For predicting response time, GPT-2 + original item text reached #7 with an RMSE of 26.073.

5.4 Data Augmentation

EDU (EduNLP) (Veeramani et al., 2024) incorporated additional data from the "Test of Narrative Language" assessment (TNL) (Fisher et al., 2019)

[5] A parameter that controls the level of randomness of the LLM output, ranging between p= 2.0 (maximum randomness) and p = 0.0 (fully deterministic).

to use in an auxiliary task. For both the shared task data and the TNL data, the authors first prompted three LLMs to annotate named entities within the data. Then, they passed each sentence with its annotated named entities as input to the LLMs, this time for the task of semantic role labeling[6]. Next, the LLMs were provided with the item, named entities, semantic roles, and the correct answer, and prompted to summarize the association between these and each answer option. The models then were instructed as follows: *"Depending on the difficulty level of the linkages between input context and [answer options], assign the input context a score in the range of 0 to 1.4"*. The best run from this approach ranked #10 for difficulty prediction (RMSE = 0.308). For modeling response time, the authors added numeric and syntactic features from LingFeat (Shaikh et al., 2022), resulting in #15 rank and an RMSE of 27.474.

SCaLARlab (Ram et al., 2024) performed data augmentation by utilizing LLMs to generate additional items with difficulty values above 0.7, to balance the training set. Three models were trained on the augmented dataset: i) BioBERT + Linguistic features as input to two different neural network architectures, ii) Word2Vec embeddings as input to various regressor models (e.g., RF, KNN, SVM), and iii) combinations of BioBERT + Linguistic features as input to the regressor models. The best run resulted in a rank of #19 for difficulty (RMSE = 0.315) and #12 for response time (RMSE = 26.945).

5.5 Linguistic features + embeddings

A number of teams experimented with combining various linguistic features with embeddings and performing model ensembling.

ITEC (Tack et al., 2024) extracted features from the Linguistic Inquiry and Word Count tool (LIWC-22) (Pennebaker et al., 2022) and TAALES 2.2 (Kyle and Crossley, 2015), which include classic linguistic features, as well as features that were not previously applied to this domain such as authenticity, clout, emotional tone, and academic vocabulary, among others. To these, the authors added Bio_ClinicalBERT embeddings (Alsentzer et al., 2019) for different combinations of item components (e.g., stem only, answer option only, etc.). These features were used as input to various regression models following feature selection and dimensionality reduction procedures. The authors also experimented with finetuning clinically pretrained BERT variations in a multi-target regression setting, as well as combining the output from all of these models into an ensemble. Best results for difficulty prediction were from RF, ranking #4 with an RMSE of 0.305, while a lasso model ranked #2 for response time prediction (RMSE = 24.116). The LWIC feature indicating the degree of "analytical thinking" for the answer options emerged as particularly noteworthy for predicting response time and, to a slightly lesser extent, difficulty.

Iran-Canada (Yousefpoori-Naeim et al., 2024) experimented with various features (including Coh-Metrix (Graesser et al., 2004) and number of medical terms) and MPNet embeddings (Song et al., 2020) as input to 15 regression models. After performing feature selection, they found that "the addition of embeddings only slightly enhances model performance", and that ensembling did not lead to major improvement. Notable features for difficulty prediction were related to cohesion, while for response time were related to length and presence of medical terms. The best run resulted in a rank of #15 for difficulty (RMSE = 0.311) and #18 for response time (RMSE = 28.714).

BC (Felice and Duran Karaoz, 2024) experimented with three approaches: i) a linear regression model using linguistic features similar to those in Ha et al. (2019), ii) several transformer models, of which RoBERTa (Liu et al., 2019) performed best, and iii) a linear regression ensemble built on the predictions of the previous two models. These systems ranked #7, #8, and #5, respectively, with an RMSE of 0.305 for the ensemble model for difficulty prediction. The BC team did not participate the response time track.

Rishikesh (Fulari and Rusert, 2024) combined embeddings from PubMedBERT-MS-MARCO (Deka et al., 2022) with linguistic features as input for a number of neural and non-neural models. The best run ranked #14 for difficulty (RMSE = 0.31) and #9 for response time (RMSE = 26.651).

BRG (Bulut et al., 2024) used Coh-Metrix features and BiomedBERT embeddings (Gu et al., 2021) within a lasso model following dimensionality reduction through PCA (Wold et al., 1987). This approach ranked #20 for predicting item difficulty (RMSE = 0.318) and #24 for response time (RMSE = 31.48).

[6] The authors also use Longformer (Beltagy et al., 2020) for named entity recognition and AllenNLP SRL (Gardner et al., 2018) for semantic role labeling.

5.6 Summary of techniques

Overall, the teams explored a wide variety of approaches, many of which performed similarly despite using different models and predictors.

Most teams experimented with all parts of the items (i.e. stem, options, correct answer), but some found different parts to be more appropriate for different tasks. The teams that used scaling were more successful, although their success cannot be solely attributed to this procedure. A variety of linguistic feature sets were explored: LWIC-22, TAALES 2.2, Coh-Metrix, SMOG, Lengths, LingFeat, as well as linguistic features from Ha et al. (2019) and Yaneva et al. (2020). The embedding types that were explored include TF-IDF, BERT, Word2Vec, Bio_ClinicalBERT, Clinical-Longformer, BERT-clinical_qa, BiomedBERT, Fastext, Bio-BERT, RoBERTa, DeBERTa, ELECTRA, MPNet, and PubMedBert-MS-MARCO. For feature engineering, the teams utilized correlation studies, multicolinearity reduction, AIC, BIC, and PCA to reduce the number of features. The modeling was performed using both traditional machine learning models (e.g., linear regression, Ridge, Lasso, ElasticNet, SGD, SVM, DT, RF, KNN, etc.) and finetuning neural models (BERT, GPT2, RoBERTa, bioBERT, XLNet, DeBERTa, DistilBERT). Customization techniques included scalar mixing, Rational Activation, multi-task learning, and a custom ANN. There was a variety of cross validation techniques: two teams used 5-folds, another two used 10-folds, and one used 5x5-fold; one team split training data into 80% and 20% training and development portions, and another split it 90% and 10% 30 times.

6 Discussion

The presented Shared Task is the first effort to benchmark the success of different methodologies on a common dataset of MCQs with known difficulties and response times. Several innovative approaches, previously unexplored in this context, were formulated. The findings are consistent with prior work, which showed that, for clinical MCQs, the prediction of item difficulty is more challenging than the prediction of response time.

6.1 Model Performance

For difficulty prediction, the models surpassed the baseline by a slight margin, with minimal variance among the solutions despite their distinct methodologies. One reason for the challenging nature of this task could be the homogeneity of the test-taker sample: the majority of questions were answered correctly by most examinees, who were highly able and motivated medical students taking the exams under high-stakes conditions as a requirement for obtaining a professional license. The models may perform differently when applied to exams targeting, for instance, K-12 students, where test-taker ability has higher variance, and difficulty distributions are more variable and less skewed. In addition, the comparable results achieved by different approaches imply multiple avenues for extracting predictive signal. An important question is whether these approaches would complement each other resulting in improved predictions.

When predicting response time, a wider variance in performance was observed, both among different models and in comparison to the baseline. A somewhat unexpected finding was the superior performance of a model solely utilizing a BERT Base model, surpassing other solutions. Another observation was the relative success of models utilizing linguistic features for predicting response time compared to their performance with predicting difficulty. Since the literature on predicting item response time is rather limited, it is not yet possible to draw inferences on how these findings compare to other exam domains.

6.2 Limitations

In formulating the shared task, we made several design choices, each contributing distinct strengths and limitations to this study.

The first decision involved utilizing proportion correct responses (known in the measurement literature as *p-values*) as the measure of item difficulty. P-values describe the interaction between an item and a sample of examinees. This sample dependency means that difficulty will only be comparable across items to the extent that the examinee samples used to calculate them are equivalent across items. (For this reason, difficulty parameters obtained using Item Response Theory (IRT) are often preferable to p-values, since they are sample independent.) A similar dependency exists for mean response time. For the data used in this shared task, examinees were randomly assigned to test forms within cohort and cohorts were reasonably stable over time making the p-values and mean response times sufficiently comparable for many expected applications.

The second design consideration was whether (and, if so, how) to rescale the target variables. Because normal distributions have many useful properties and most parametric tests make a normality assumption of one kind or another, it is not uncommon to transform data such that they approximate a normal distribution. For proportion correct, a logit transformation often accomplishes this; and for response times, a log transformation is typical. Such transformations will be familiar to researchers accustomed to working with these kinds of data and for many applications transformations like this are justified and sensible. Nevertheless, because there are other occasions when it may be preferable to keep values on their original scale, it is necessary to carefully consider an intended application for a dataset before deciding how it should be rescaled.

For example, when RMSE is used to evaluate predicted values—as it was for this shared task—nonlinear transformations have the effect of weighting errors differently depending on the values of the predictions and the target variables. Under these conditions, applying a logit transformation to proportion correct values would have the effect of weighting errors for values nearer to 1 or nearer to 0 more than the errors for values nearer to .5. While this may be desirable for certain applications, here we choose to leave the question of application open and weight all errors equally. To this end, only a linear transformation was applied to the proportion correct values and mean response times were left untransformed. Participants were, of course, free to transform the data in any manner they deemed helpful provided their predictions were submitted on the scale of the original values.

Third, the data for this task was limited to clinical MCQs, limiting the inferences that can be made about the generalizability of these methodologies to other domains. How the approaches generalize is an empirical question, however, one can speculate that they might be less effective in a math examination where items often contain minimal text, and more beneficial in reading-comprehension examinations where the text's complexity may be deliberately varied to manipulate difficulty. In an ideal world, future shared tasks on this topic should span multiple content domains and examinee populations with different characteristics, while remaining equally rigorous in terms of the conditions under which the examinee responses were collected.

6.3 Ethical Considerations

The data used in the Shared Task were obtained with the explicit permission of the data and copyright owners for the purposes of the Shared Task. Beyond this competition, the data are available upon request, following a data use agreement intended to ensure, to the extent possible, its ethical use in research. Test taker responses were used in aggregate, such that it is not possible to trace responses to individual examinees.

6.4 Impact

While benchmarking and fostering novel methodologies is a key contribution of this Shared Task, its impact reaches further. The competition spurred the development of a body of research on modeling item response time, a considerably less explored area. Moreover, many solutions were not narrowly tailored to the clinical realm and are potentially applicable to diverse domains and datasets. Further still, it is notable that the significance of these studies is not limited to the field of education—difficulty assessment beyond mere readability is an exciting frontier with implications for cognition and machine comprehension.

7 Conclusion

The First Shared Task on Automated Prediction of Difficulty and Response Time featured a set of 667 MCQs from a high-stakes clinical exam. Seventeen teams submitted solutions and twelve teams submitted system report papers. For Track 1, Item Difficulty Prediction, the best-performing solution achieved an RMSE of 0.299 compared to the DummyRegressor baseline of 0.311. For Track 2, Response Time Prediction, the best solution achieved an RMSE of 23.927 compared to 31.68 for the baseline. The paper summarized the methodologies proposed by the participants and discussed the contributions and limitations of the competition.

Despite the progress made, the challenge of predicting item characteristics remains formidable. Meeting this challenge necessitates not only the continued development of innovative methodologies but also the establishment of shared resources, such as public datasets containing reliable parameter estimates across various domains. Such efforts will facilitate cross-domain evaluation, fostering a more comprehensive understanding of the underlying mechanisms driving item difficulty and response time prediction.

References

Samah AlKhuzaey, Floriana Grasso, Terry R Payne, and Valentina Tamma. 2023. Text-based Question Difficulty Prediction: A Systematic Review of Automatic Approaches. *International Journal of Artificial Intelligence in Education*, pages 1–53.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. 2023. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*.

Tahani Alsubait, Bijan Parsia, and Ulrike Sattler. 2013. A similarity-based theory of controlling mcq difficulty. In *e-Learning and e-Technologies in Education (ICEEE), 2013 Second International Conference on*, pages 283–288. IEEE.

Peter Baldwin, Victoria Yaneva, Janet Mee, Brian E Clauser, and Le An Ha. 2020. Using natural language processing to predict item response times and improve test construction. *Journal of Educational Measurement*.

Lisa Beinborn, Torsten Zesch, and Iryna Gurevych. 2014. Predicting the difficulty of language proficiency tests. *Transactions of the Association for Computational Linguistics*, 2:517–530.

Lisa Beinborn, Torsten Zesch, and Iryna Gurevych. 2015. Candidate evaluation strategies for improved difficulty prediction of language tests. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–11.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Luca Benedetto, Giovanni Aradelli, Paolo Cremonesi, Andrea Cappelli, Andrea Giussani, and Roberto Turrin. 2021. On the application of Transformers for estimating the difficulty of Multiple-Choice Questions from text. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 147–157, Online. Association for Computational Linguistics.

Luca Benedetto, Andrea Cappelli, Roberto Turrin, and Paolo Cremonesi. 2020. Introducing a Framework to Assess Newly Created Questions with Natural Language Processing. In *Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part I*, page 43–54, Berlin, Heidelberg. Springer-Verlag.

Sheng Bi, Xiya Cheng, Yuan-Fang Li, Lizhen Qu, Shirong Shen, Guilin Qi, Lu Pan, and Yinlin Jiang. 2021. Simple or complex? complexity-controllable question generation with soft templates and deep mixture of experts model. *arXiv preprint arXiv:2110.06560*.

Okan Bulut, Guher Gorgun, and Bin Tan. 2024. Item Difficulty and Response Time Prediction with Large Language Models: An Empirical Analysis of USMLE Items. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Zeming Chen, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, et al. 2023. Meditron-70b: Scaling medical pretraining for large language models. *arXiv preprint arXiv:2311.16079*.

PRITAM Deka, ANNA Jurek-Loughrey, and P Deepak. 2022. Improved methods to aid unsupervised evidence-based fact checking for online heath news. *J. Data Intell.*, 3(4):474–504.

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*.

George Dueñas, Sergio Jimenez, and Geral Eduardo Mateus Ferro. 2024. UPN-ICC at BEA 2024 Shared Task: Leveraging LLMs for Multiple-Choice Questions Difficulty Prediction. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Yasmine H El Masri, Steve Ferrara, Peter W Foltz, and Jo-Anne Baird. 2017. Predicting item difficulty of science national curriculum tests: The case of key stage 2 assessments. *The Curriculum Journal*, 28(1):59–82.

Mariano Felice and Zeynep Duran Karaoz. 2024. The British Council submission to the BEA 2024 shared task. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Evelyn L Fisher, Andrea Barton-Hulsey, Casy Walters, Rose A Sevcik, and Robin Morris. 2019. Executive functioning and narrative language in children with dyslexia. *American journal of speech-language pathology*, 28(3):1127–1138.

Roy Freedle and Irene Kostin. 1993. The prediction of TOEFL reading item difficulty: implications for construct validity. *Language Testing*, 10(2):133–170.

Rishikesh Fulari and Jon Rusert. 2024. Utilizing Machine Learning to Forecast Question Difficulty and Response. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Gao, Lidong Bing, Wang Chen, Michael R Lyu, and Irwin King. 2018. Difficulty controllable generation of reading comprehension questions. *arXiv preprint arXiv:1807.03586*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. *arXiv preprint arXiv:1803.07640*.

Sebastian Gombert, Lukas Menzel, and Hendrik Drachsler. 2024. Predicting Item Difficulty and Item Response Time with Scalar-mixed Transformer Encoder Models and Rational Network Regression Heads. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Arthur C Graesser, Danielle S McNamara, Max M Louwerse, and Zhiqiang Cai. 2004. Coh-metrix: Analysis of text on cohesion and language. *Behavior research methods, instruments, & computers*, 36(2):193–202.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

Le An Ha, Victoria Yaneva, Peter Baldwin, and Janet Mee. 2019. Predicting the difficulty of multiple choice questions in a high-stakes medical exam. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 11–20, Florence, Italy. Association for Computational Linguistics.

Perry N. Halkitis et al. 1996. Estimating Testing Time: The Effects of Item Characteristics on Response Latency. *ERIC*.

Jun He, Li Peng, Bo Sun, Lejun Yu, and Yinghui Zhang. 2021. Automatically predict question difficulty for reading comprehension exercises. In *2021 ieee 33rd international conference on tools with artificial intelligence (ictai)*, pages 1398–1402. IEEE.

Fu-Yuan Hsu, Hahn-Ming Lee, Tao-Hsing Chang, and Yao-Ting Sung. 2018. Automated estimation of item difficulty for multiple-choice tests: An application of word embedding techniques. *Inf. Process. Manag.*, 54:969–984.

Zhenya Huang, Qi Liu, Enhong Chen, Hongke Zhao, Mingyong Gao, Si Wei, Yu Su, and Guoping Hu. 2017. Question difficulty prediction for reading problems in standard tests. In *AAAI*, pages 1352–1359.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Ghader Kurdi, Bijan Parsia, and Uli Sattler. 2016. An experimental evaluation of automatically generated multiple choice questions from ontologies. In *OWL: Experiences And directions–reasoner evaluation*, pages 24–39. Springer.

Kristopher Kyle and Scott A Crossley. 2015. Automatically assessing lexical sophistication: Indices, tools, findings, and application. *Tesol Quarterly*, 49(4):757–786.

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*.

Ekaterina Loginova, Luca Benedetto, Dries Benoit, and Paolo Cremonesi. 2021. Towards the application of calibrated transformers to the unsupervised estimation of question difficulty from text. In *RANLP 2021*, pages 846–855. INCOMA.

Anastassia Loukina, Su-Youn Yoon, Jennifer Sakano, Youhua Wei, and Kathy Sheehan. 2016. Textual complexity as a predictor of difficulty of listening items in language proficiency tests. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3245–3253.

Arya D McCarthy, Kevin P Yancey, Geoffrey T LaFlair, Jesse Egbert, Manqian Liao, and Burr Settles. 2021. Jump-starting item parameters for adaptive language tests. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 883–899.

Robert J Mislevy. 1988. Exploiting auxiliary information about items in the estimation of rasch item difficulty parameters. *Applied Psychological Measurement*, 12(3):281–296.

Cynthia G. Parshall et al. 1994. Response Latency: An Investigation into Determinants of Item-Level Timing. *ERIC*.

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.

JW Pennebaker, RL Boyd, RJ Booth, A Ashokkumar, and ME Francis. 2022. Linguistic inquiry and word count: Liwc-22. pennebaker conglomerates.

Kyle Perkins, Lalit Gupta, and Ravi Tammana. 1995. Predicting item difficulty in a reading comprehension test with an artificial neural network. *Language Testing*, 12:34 – 53.

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.

Gummuluri Venkata Ravi Ram, Kesanam Ashinee, and Anand Kumar M. 2024. Leveraging Physical and Semantic Features of text item for Difficulty and Response Time Prediction of USMLE Questions. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Alvaro Rodrigo, Sergio Moreno-Álvarez, and Anselmo Peñas. 2024. UNED team at BEA 2024 Shared Task: Testing different Input Formats for predicting Item Difficulty and Response Time in Medical Exams. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Ana-Cristina Rogoz and Radu Tudor Ionescu. 2024. UnibucLLM: Harnessing LLMs for Automated Prediction of Item Difficulty and Item Response Time. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Stefan Schneiderand, Haomiao Jin, Bart Orriens, Doerte U. Junghaenel, Arie Kapteyn, Erik Meijer, and Arthur A. Stone. 2023. Using Attributes of Survey Items to Predict Response Times May Benefit Survey Research. *Field Methods*, 35:87–99.

Burr Settles, Geoffrey T. LaFlair, and Masato Hagiwara. 2020. Machine learning–driven language assessment. *Transactions of the Association for computational Linguistics*, 8:247–263.

Samira Shaikh, Thiago Ferreira, and Amanda Stent, editors. 2022. *Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges*. Association for Computational Linguistics, Waterville, Maine, USA and virtual meeting.

Russell Winsor Smith. 2000. *An exploratory analysis of item parameters and characteristics that influence item level response time*. The University of Nebraska-Lincoln.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pretraining for language understanding. *Advances in neural information processing systems*, 33:16857–16867.

Lisa Ann Keller Stowe. 2002. *Small-sample item parameter estimation in the three parameter logistic model: Using collateral information*. Ph.D. thesis, University of Massachusetts Amherst.

Yunik Susanti, Takenobu Tokunaga, Hitoshi Nishikawa, and Hiroyuki Obari. 2017. Controlling item difficulty for automatic vocabulary question generation. *Research and Practice in Technology Enhanced Learning*, 12.

Hariharan Swaminathan, Ronald K Hambleton, Stephen G Sireci, Dehui Xing, and Saba M Rizavi. 2003. Small sample estimation in dichotomous item response models: Effect of priors based on judgmental information on the accuracy of item parameter estimates. *Applied psychological measurement*, 27(1):27–51.

David B. Swanson, Susan M. Case, Douglas R. Ripkey, Brian E. Clauser, and Matthew C. Holtman. 2001. Relationships Among Item Characteristics, Examine Characteristics, and Response Times on USMLE Step 1. *Academic Medicine*, 76:114–116.

Anaïs Tack, Siem Buseyne, Changsheng Chen, Robbe D'hondt, Michiel De Vrindt, Alireza Gharahighehi, Sameh Metwaly, Felipe Kenji Nakano, and Ann-Sophie Noreillie. 2024. ITEC at BEA 2024 Shared Task: Predicting Difficulty and Response Time of Medical Exam Questions with Statistical Machine Learning and Language Models. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Hariram Veeramani, Natarajan Balaji Shankar Balaji, and Surendrabikram Thapa. 2024. Large Language Model-based Framework for Item Difficulty and Response Time Estimation for Assessments. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52.

Kang Xue, Victoria Yaneva, Christopher Runyon, and Peter Baldwin. 2020. Predicting the difficulty and response time of multiple choice questions using transfer learning. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 193–197, Seattle, WA, USA → Online. Association for Computational Linguistics.

Victoria Yaneva, Peter Baldwin, Christopher Runyon, et al. 2023. Extracting linguistic signal from item text and its application to modeling item characteristics. In *Advancing Natural Language Processing in Educational Assessment*, pages 167–182. Routledge.

Victoria Yaneva, Le An Ha, Peter Baldwin, and Janet Mee. 2020. Predicting item survival for multiple choice questions in a high-stakesmedical exam. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020) , Marseille, 11–16 May 2020*, page 6814-6820.

Victoria Yaneva, Daniel Jurich, Peter Baldwin, et al. 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.

Mehrdad Yousefpoori-Naeim, Shayan Zargari, and Zahra Hatami. 2024. Using machine learning to predict item difficulty and response time in medical tests. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Ya Zhou and Can Tao. 2020. Multi-task BERT for problem difficulty prediction. In *2020 International Conference on Communications, Information System and Computer Engineering (CISCE)*, pages 213–216.

Predicting Item Difficulty and Item Response Time with Scalar-mixed Transformer Encoder Models and Rational Network Regression Heads

Sebastian Gombert[1], **Lukas Menzel**[2], **Daniele Di Mitri**[1], **and Hendrik Drachsler**[1,2,3,4]
[1]DIPF: Leibniz Institute for Research and Information in Education, Frankfurt, Germany
[2]studiumdigitale & [3]Computer Science Department, Goethe University Frankfurt, Germany
[4]Department of Online Learning and Instruction, Open University, Heerlen, Netherlands
{s.gombert,d.dimitri,h.drachsler}@dipf.de
menzel@sd.uni-frankfurt.de

Abstract

This paper describes a contribution to the *BEA 2024 Shared Task on Automated Prediction of Item Difficulty and Response Time*. The participants in this shared task are to develop models for predicting the difficulty and response time of multiple-choice items in the medical field. These items were taken from the United States Medical Licensing Examination® (USMLE®), a high-stakes medical exam. For this purpose, we evaluated multiple BERT-like pre-trained transformer encoder models, which we combined with Scalar Mixing and two custom 2-layer classification heads using learnable Rational Activations as an activation function, each for predicting one of the two variables of interest in a multi-task setup. Our best models placed first out of 43 for predicting item difficulty and fifth out of 34 for predicting Item Response Time.

1 Introduction

According to Madaus and Airasian (1970), assessments are arguably among the core components of education. They help diagnose and monitor learners' skill levels and, thus, function as a basis for downstream educational decisions. Depending on their concrete function, they can be further categorized. Placement assessments are needed to recommend courses for learners at an appropriate level. Formative assessments are required to monitor learning progress. Summative assessments are needed to measure learners' outcomes.

Each assessment comprises multiple items, i.e., individual tasks test-takers must complete. For standardized assessments, items must be evaluated to guarantee fair and comparable outcomes. In this context, multiple factors must be assessed as listed in the *Standards for educational and psychological testing* (Association et al., 1985).

Among these factors are *Item Difficulty*, i.e., a numerical variable describing the overall difficulty of solving a given item, and *Item Response Time*, which encodes the overall time needed to solve an item measured in seconds. Traditionally, *Item Difficulty* has been assessed using methods such as *Rasch Analysis* (Rasch, 1960) or *Item Response Theory* (An and Yung, 2014). Both of them rely on collection data from pre-evaluations with cohorts of test takers. As administering respective pre-evaluation steps is still a labour-intensive and costly process (Settles et al., 2020), there has been ongoing research on automating these procedures using machine learning methods with a higher potential for generalization.

One of these instances is the *First Shared Task on Automated Prediction of Difficulty and Response Time for Multiple Choice Questions* (Yaneva et al., 2024). In this paper, we describe a submission to this shared task, which placed first for predicting *Item Difficulty* and fifth for predicting *Item Response Time*.

2 Related Work

Both the prediction of *Item Difficulty* and *Item Response Time* for multiple choice questions utilizing natural language processing are comparably novel tasks. Earlier research on predicting *Item Difficulty* tackled mostly other item formats such as C-tests, a form of fill-in-the-blank test aimed at testing language proficiency, (Beinborn et al., 2015) or constructed response items (Padó, 2017). In the context of language learning, Settles et al. (2020) developed a method to assess the difficulty of various types of items for language learning in terms of the *CEFR* framework.

Early research on predicting *Item Difficulty* for multiple choice questions was conducted by Ha et al. (2019), who fit various feature-based models using heterogeneous sets of features incorporating embeddings, as well as lexical, syntactic, semantic, cohesion-based, and psycholinguistic features to predict *Item Difficulty* for a large-scale dataset comprised of *United States Medical Licensing Ex-*

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 483–492
June 20, 2024 ©2024 Association for Computational Linguistics

amination® (USMLE®) items. The authors also use features derived from information retrieval systems. They reason that retrieving an answer for a given question through Information Retrieval might predict the difficulty of cognitively retrieving an answer. Subsequent work by Yaneva et al. (2020) and Yaneva et al. (2021) used similar approaches to predict *Item Survival* and *Item Response Complexity*.

For predicting *Item Response Time*, Baldwin et al. (2021) used feature-based models using primarily the same features and algorithms which Ha et al. (2019) applied for predicting *Item Difficulty*. They found that embeddings and linguistic features were robust in predicting *Item Response Time*, with IR-based features being less predictive while still holding some degree of predictive power. Yaneva et al. (2023) combined linguistic features with static embeddings produced by word2vec and contextual word embeddings produced by non-fine-tuned BERT models to predict a range of item characteristics, including *Item Response Time*.

What becomes apparent when reviewing the past literature on the topic is that transformer-encoder language models such as BERT (Devlin et al., 2019) have not been fine-tuned for the prediction of *Item Difficulty* and *Item Response Time* as of now. This can be regarded as a clear research gap, given that transformer encoders could push the state of the art for a wide range of tasks in natural language processing and outperformed more traditional feature-based approaches for these (Rogers et al., 2020).

3 Method

To close this gap, we aim to evaluate the overall predictiveness of pre-trained transformer encoder language models for *Item Difficulty* and *Item Response Time* in our submission for the *First Shared Task on Automated Prediction of Difficulty and Response Time for Multiple Choice Questions* (Yaneva et al., 2024).

3.1 Dataset

The dataset used for this task was provided by Yaneva et al. (2024) and consists of multiple choice items that were previously used for the *United States Medical Licensing Examination® (USMLE®)*. It is divided into a training and a test set, with the training set comprising 466 and the test set comprising 201 items. Each item consists of a prompt with up to 10 different response options, of which a single one is correct. Moreover, for each item, it is remarked whether the response options come in the form of texts or images (the images are not provided with the dataset; instead, there are descriptions of what is depicted) and if the items belong to the first, second or third step of the *USMLE®*.

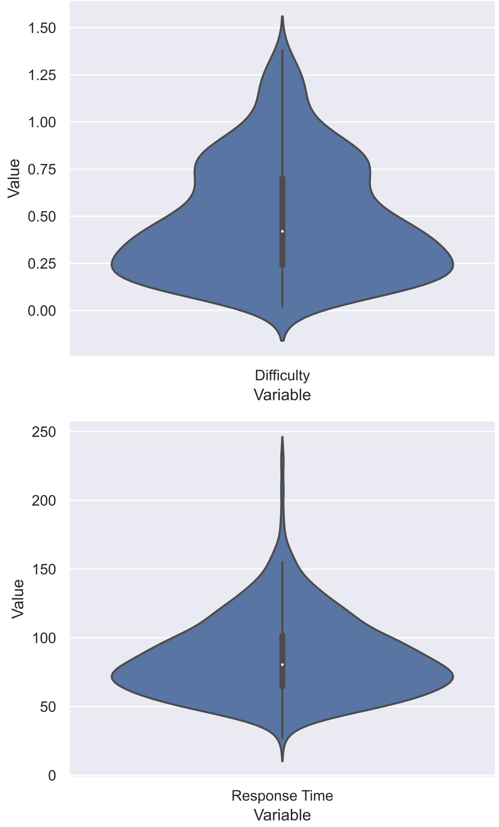


Figure 1: Violinplots depicting the general distributional properties of *Item Difficulty* and *Item Response Time*.

Each item is given a single rating for *Item Difficulty* and one for *Item Response Time*. Figure 1 shows the distribution of values for both of them. Going by *Shapiro-Wilk*, neither *Item Difficulty* ($W = 0.93, p < 0.000$) nor *Item Response Time* ($W = 0.94, p < 0.000$) follow a normal distribution. However, as Figure 2 reveals, both are significantly correlated, which is also confirmed by *Pearson's* ($r = 0.49, p < 0.000$) and *Spearman's* ($r = 0.52, p < 0.000$) correlation coefficients.

As the difficulty of an item very likely influences the time needed to think about the correct answer, it can be speculated that there is, to a certain degree, a causal relationship between both variables. However, given that the r values are not higher, it can also be concluded that this is not the only factor influencing the exact outcome of both variables for each item.

3.2 System Description

The architecture we implemented for this shared task is derived from the modified transformer-based model implemented by Gombert et al. (2022) for automated short answer scoring, where it outperformed regular transformer-based models for this task. Our architecture can be flexibly applied to various regression and classification tasks. It is a deep neural network architecture based upon regular *BERT*-like transformer-encoder language models (Devlin et al., 2019). The typical BERT regression setup uses a single output neuron. This neuron is fed with the last layer's classification token output. Our setup, however, is modified.

The first difference to the standard BERT implementation is the usage of scalar mixing. Scalar mixing calculates a weighted mean of all hidden layers of a transformer language model. The weights from which this mean is calculated are fit during training. This technique was mainly applied to investigate the influence of different pre-trained layers on a given prediction (Tenney et al., 2019; Kuznetsov and Gurevych, 2020). Still, it can also be used as a regular neural network building block.

Different layers of BERT-like models learn representations for different linguistic phenomena (Tenney et al., 2019). Using scalar mixing lets us exploit all these representations, instead of only the output of the last layer, while simultaneously learning their importance for the final output. Scalar mixing can be depicted using the following equation with tensors $t_1, ..., t_n$ being the hidden layer outputs, and γ and $w_1, ..., w_n$ being the learnable parameters:

$$S(t_1, ..., t_n) = \gamma \sum_{j=0}^{n} softmax(w_j) t_j \quad (1)$$

The second adjustment to the classification heads is to use a two-layer setup. The output of the intermediate layer runs through a *Rational Activation* (Molina et al., 2020), a form of learnable activation function whose shape is optimized during training; thus, a "Rational Network". This activation function outperformed non-learnable activation functions for multiple architectures and benchmarks. Rational Activations are based upon Padé approximants (Brezinski et al., 1995), which can generally be optimized to approximate various functions, including typical activation functions. Given a hypothetical optimal activation function $f(x)$ for a problem at hand, one can approximate this function by learning a Padé approximant $F(x)$ of the pre-defined orders n and m using the following equation where coefficients a_j and a_k are learned during training:

$$F(x) = \frac{\sum_{j=0}^{m} a_j x^j}{1 + |(\sum_{k=1}^{m}) a_k x^k|} \quad (2)$$

Another important aspect of our model is the use of multi-task learning. As Peng et al. (2020) put it, "[m]ulti-task learning (MTL) is a field of machine learning where multiple tasks are learned in parallel while using a shared representation", with "representation" referring to the internal embeddings put out by the different model layers. Although the shared task rules prevented using one of the two variables to predict the other directly, they did not prevent implementing a system simultaneously predicting both. As shown in section 3, *Item Difficulty* and *Item Response Time* are significantly correlated in the training set. While this does not necessarily prove a causal relationship, it implies that the internal representations used to predict one of the two variables can likely benefit the prediction of the other. Therefore, using shared representations will likely lead to improved predictions for both variables.

Multi-task learning is usually conducted by attaching multiple prediction heads to the base model for transformer-encoder models. Our setup involves the usage of a complete distinct regression head per variable, each with separate units for *Scalar Mixing* and *Rational Activations*, and distinct linear layers. We reason, while the transformer encoder learns shared representations during fine-tuning, both variables might require a stronger or weaker emphasis on different model layers during *Scalar Mixing*. Moreover, an optimal learned activation function $F(x)$ might look different for both.

Given an item k, the model receives the following corresponding input $I(k)$, with $\oplus$ referring to the separation token of a given model, $s_k \in \{1, 2, 3\}$ to the exam step, $t_k \in \{TEXT, PIX\}$ to the item type, p_k to the item prompt, $r_{k1}, ..., r_{kn}$ to the possible answers, and $c_k \in \{r_{k1}, ..., r_{kn}\}$ to the correct answer:

$$I(k) = s_k \oplus t_k \oplus p_k \oplus r_{k1} \oplus ... \oplus r_{kn} \oplus c_k \quad (3)$$

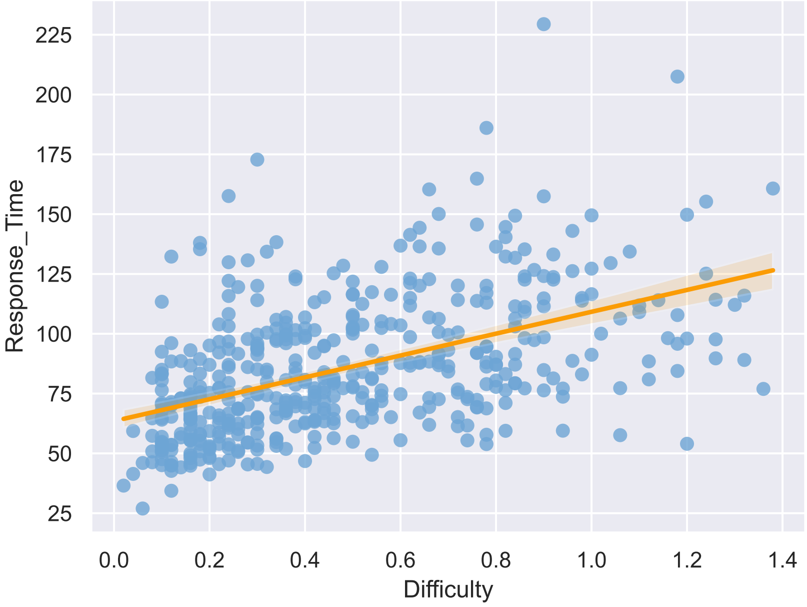


Figure 2: This scatterplot depicts the relationship between *Item Difficulty* and *Item Response Time*.

During training, the mean loss for both variables is calculated to acquire the gradients for backpropagation. Since *Item Response Time* and *Item Difficulty* are on different scales, a naïve approach would strongly bias the model towards *Item Response Time*. For this reason, we divide *Item Response Time* by 100 to get similar scales for both variables. Consequently, the model outputs for *Item Response Time* must be multiplied by 100 again to acquire the actual item response time. With a model $M(x)$ receiving an input as defined by $I(k)$, v_k being the *Item Difficulty*, and w_k being the *Item Response Time* of k, the following equation illustrates this:

$$M(I(k)) = (v_k, \frac{w_k}{100}) \tag{4}$$

Figure 3 illustrates the overall system setup.

3.3 Evaluation

3.3.1 Pre-Evaluation (Model Selection)

In a pre-evaluation step, we aimed to select the most appropriate transformer language model to use as the basis for our shared task submission. Therefore, we evaluated the architecture described in the System Description section with different pre-trained transformer-encoder language models. All models were implemented using the *Huggingface Transformers* framework (Wolf et al., 2020). However, we implemented our own training and evaluation procedures. These are the following models:

- BERT-large[1]: this model is the original BERT model as described in Devlin et al. (2019).

- RoBERTa-large[2]: This model is an established BERT variant that was pre-trained on a larger data set without the usage of next sentence prediction and outperforms regular BERT on established benchmarks such as *SuperGLUE* (Wang et al., 2019).

- ELCTRA-large[3]: this model was published by Clark et al. (2020). Unlike BERT and RoBERTa, it is pre-trained in an adversarial setup using two models that implement a variation of masked language modelling. One model, the generator, predicts masked tokens. The other model, the discriminator, then must classify random input tokens concerning whether they were generated or ground truth.

- DeBERTa-v3-large[4]: this model was published by He et al. (2023). It uses *disentangled attention* to separately encode the content and

[1] https://huggingface.co/google-bert/bert-large-uncased

[2] https://huggingface.co/FacebookAI/roberta-large

[3] https://huggingface.co/google/electra-large-discriminator

[4] https://huggingface.co/microsoft/deberta-v3-large

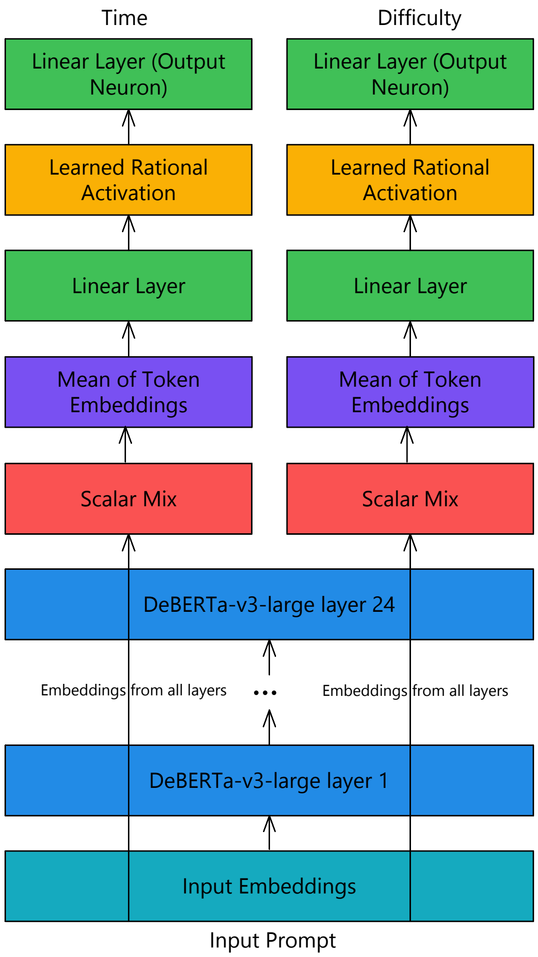

Time
Difficulty
Linear Layer (Output Neuron)
Linear Layer (Output Neuron)
Learned Rational Activation
Learned Rational Activation
Linear Layer
Linear Layer
Mean of Token Embeddings
Mean of Token Embeddings
Scalar Mix
Scalar Mix
DeBERTa-v3-large layer 24
Embeddings from all layers
...
Embeddings from all layers
DeBERTa-v3-large layer 1
Input Embeddings
Input Prompt

Figure 3: This diagram depicts the general architecture of our models. A given input is encoded into static embeddings. These are then propagated through all layers of a given pre-trained transformer encoder language model. The static embeddings and the outputs of all layers are propagated into the respective scalar mixing units, where a weighted mean is calculated from the individual tensors per variable. These are then propagated into the individual regression heads.

position of a token within an input text. Moreover, it is pre-trained using a specialized adversarial setup similar to ELECTRA. We chose this model since it is the best-performing open BERT-like model on the *SuperGLUE* (Wang et al., 2019) leaderboard[5].

- BiomedBERT-large[6]: This model is a BERT variant which was published by Tinn et al. (2023). It is trained identically to BERT but uses biomedical data exclusively (abstracts crawled from PubMed). We evaluated this model for the shared task since its dataset also stems from the biomedical domain.

- BiomedELECTRA-large[7]: This model is an ELECTRA variant which was published by Tinn et al. (2023). It is trained identically to ELECTRA but uses biomedical data exclusively (abstracts crawled from PubMed). We evaluated this model for the shared task since its dataset also stems from the biomedical domain.

We also added two simpler baseline models. We used *Linear Regression* and *Random Forests* as algorithms, which both are given the following features:

- *Tf-ifd*-encoded character trigrams for the item prompt and each answer option, motivated by the fact that character n-gram frequencies can provide valuable signals in terms of predicting readability (Imperial and Kochmar, 2023), which should be correlated with *Item Difficulty* and *Item Response Time*, given the results from Ha et al. (2019) and Baldwin et al. (2021).

- The overall number of tokens of the item prompt, motivated by the general observation of text length being correlated to text complexity as reported by DuBay (2007).

Additionally, we added dummy regressors that consistently predict the respective mean.

The evaluation was conducted solely on the training set using 5x5 cross-validation implemented via the *RepeatedKFold* class from *Scikit-learn* (Pedregosa et al., 2011). We trained for four epochs and reported the best results achieved during one of these epochs. All runs used the same random seed, namely *1*, to keep the results perfectly comparable. For each model, we measured *RMSE* (the primary evaluation metric of the shared task), *MAE* and r. To this, we added r_s to measure to which degree the models can correctly rank the items by the predicted variables without explicitly considering the exact predictions. Table 1 shows the respective results, ranked by *RMSE*.

It is visible that the correct prediction of the *Item Difficulty* is nearly impossible using our proposed method with the given data. None of the models

[5] https://super.gluebenchmark.com/leaderboard

[6] https://huggingface.co/microsoft/BiomedNLP-BiomedBERT-large-uncased-abstract

[7] https://huggingface.co/microsoft/BiomedNLP-BiomedELECTRA-large-uncased-abstract

Item Difficulty				
Model	RMSE ↓	MAE	r	r_s
ELECTRA	**0.31**	**0.25**	**0.19**	**0.16**
RoBERTa	**0.31**	**0.25**	0.17	**0.16**
DeBERTa-v3	**0.31**	0.26	0.17	0.15
Dummy (Mean)	***0.31***	*0.26*	-	-
Random Forests	***0.31***	*0.26*	*0.09*	*0.07*
BERT	0.32	0.27	0.16	0.14
BiomedBERT	0.32	0.26	0.11	0.11
Linear Regression	*0.32*	*0.26*	*0.11*	*0.07*
BiomedELECTRA	0.33	0.27	0.12	0.10
Item Response Time				
Model	RMSE ↓	MAE	r	r_s
DeBERTa-v3	**23.05**	**17.48**	**0.63**	**0.65**
BERT	23.52	17.76	0.60	0.64
RoBERTa	23.76	17.79	0.61	0.64
BiomedELECTRA	23.88	17.87	0.61	0.63
BiomedBERT	23.97	18.02	0.59	0.62
ELECTRA	24.68	18.57	0.60	0.64
Dummy (Mean)	*46.87*	*37.77*	-	-
Random Forests	*47.13*	*38.56*	*0.19*	*0.22*
Linear Regression	*47.60*	*38.87*	*0.17*	*0.17*

Table 1: The results of our pre-evaluation experiments to determine the strongest models ranked by RMSE. All results were calculated during 5x5 cross-validation runs.

Item Difficulty					
Model	RMSE ↓	MAE	r	r_s	Rank
ELECTRA	**0.29**	**0.24**	**0.27**	**0.25**	**1/43**
RoBERTa	0.30	0.24	0.24	0.20	3/43
Dummy	*0.31*	-	-	-	16/43
DeBERTa-v3	0.31	0.25	0.21	0.19	17/43
Item Response Time					
Model	RMSE ↓	MAE	r	r_s	Rank
UNED run2	**23.92**	-	-	-	**1/34**
RoBERTa	25.64	17.94	0.60	0.67	5/34
ELECTRA	25.87	19.14	0.57	0.65	6/34
DeBERTa-v3	27.30	21.48	0.56	0.63	14/34
Dummy	*31.68*	-	-	-	25/34

Table 2: The final shared task evaluation results. For *Item Difficulty*, we report the results of our models and the baseline dummy model of the shared task organizers. For *Item Response Time*, we also report the results of the overall winning system from a competing team called *UNED run2*.

we tested achieved a better *RMSE* score than the dummy regressor, meaning the models hold almost no predictive power. The model based on *BioMED-BERT-large* and the *Linear Regression* baseline are outperformed by this dummy regressor in terms of *RMSE*. Nonetheless, the r and r_s results show that all transformer-based models are at least more successful in modelling the *Item Difficulty* than the baselines. However, this success is still minimal.

Our pre-evaluations yielded better results for *Item Response Time*. Here, all transformer-based models significantly outperformed the baseline models. This means it is possible – to a certain degree – to model *Item Response Time* with our proposed method and the given data. While models based on *BioMED-BERT-large* and *DeBERTa-v3-large* achieve a similar *RMSE*, the model based on DeBERTa-v3-large outperforms all other models in terms of r and r_s, meaning it is the overall best model.

3.3.2 Shared Task Evaluation

The shared task organizers allowed the submission of up to three predictions per variable. We submitted results predicted with models based upon *ELECTRA*, *RoBERTa* and *DeBERTa-v3*. *BERT*, *Biomed-BERT* and *Biomed-ELECTRA* were not used since they performed worse for the prediction of the *Item Difficulty* while achieving very similar results to the other models for the *Item Response Time*. For this purpose, all three models were re-trained on the whole training set for four epochs. While the models based upon *ELECTRA* and *RoBERTa* achieved very high placements on the shared task leaderboard for both variables, the model based on *DeBERTa-v3* performed worse, which is a surprising outcome.

The overall trends observed during our pre-evaluation steps continued into the final shared task evaluations. While for the *Item Difficulty*, barely any system could show a performance superior to a dummy regressor baseline, the *Item Response Time* was easier to predict. Interestingly, the model based on *DeBERTa-v3* ranks the worst out of our models for both variables despite being the best-performing approach for predicting the *Item Response Time* during the pre-evaluations. However, except for this, the results line up.

Going by r and r_s, it is visible that predictions and ground truth values are positively correlated for both variables. However, a trend that is observable for all models and both variables is revealed in Figure 4. On average, the predicted values are lower than the ground truth. This pattern is more drastic for the *Item Difficulty* but also visible for the *Item Response Time*.

4 Discussion

The research at hand has multiple implications. First, we proved that using established pre-trained transformer-encoder language models for predicting the *Item Difficulty* and the *Item Response Time* can be a viable choice overall. Moreover, we could also show that our adjustments to the typical BERT

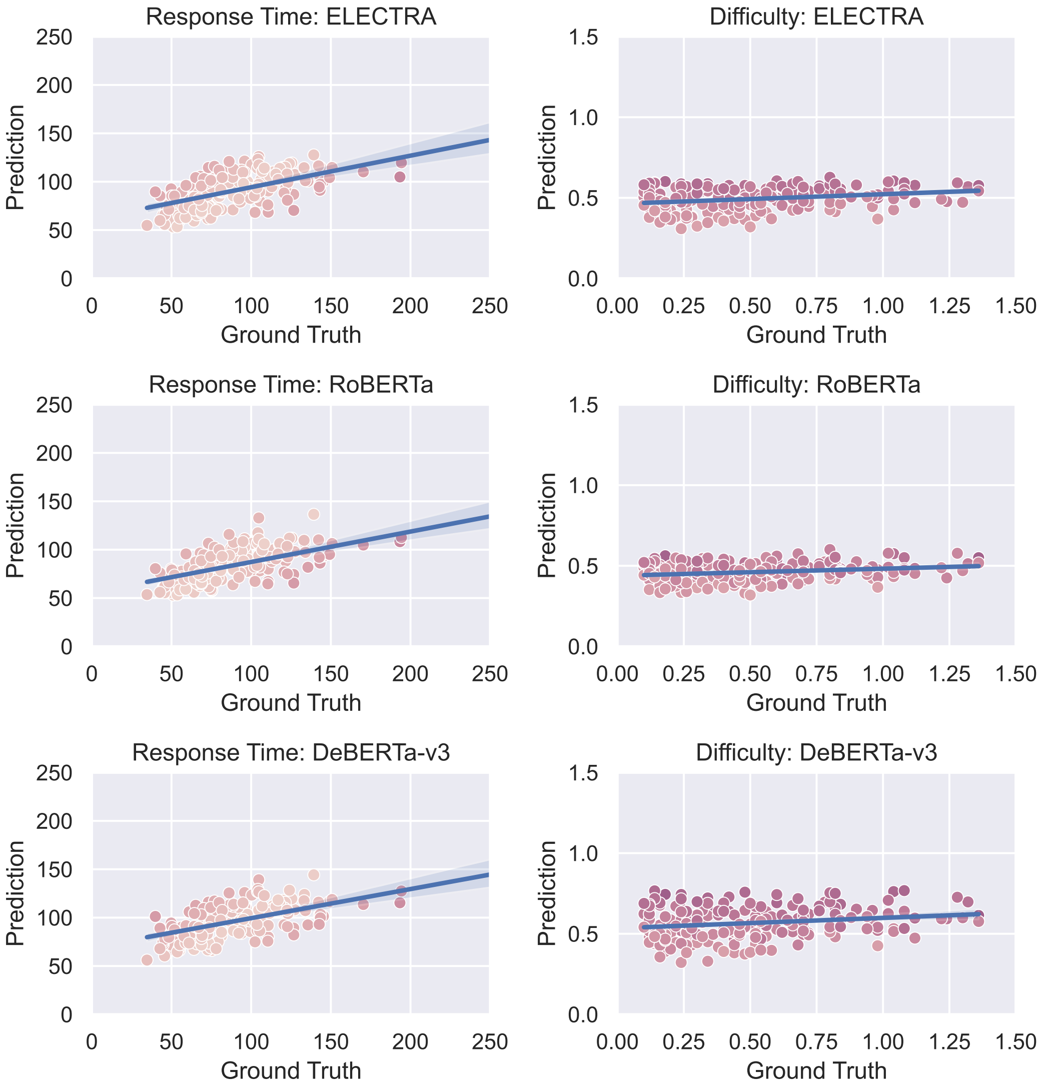


Figure 4: These regression plots illustrate the relationship between ground truth values and model predictions. The x-axis refers to the ground truth value of a given data point, while the y-axis refers to the respective predicted value. The individual data points' colour coding indicates the differences between ground truth and prediction, with a darker colour indicating a larger difference.

architecture proved fruitful. These adjustments let our models achieve very competitive performance in the shared task, with our best model even winning one of the two tracks (*Item Difficulty*).

In theory, our approach can be easily integrated with the past feature-based models published by Ha et al. (2019), Yaneva et al. (2020), Yaneva et al. (2021) and Baldwin et al. (2021). For this purpose, one needs to fine-tune a respective model. One can then use the output of all intermediate layers as embeddings. Using an algorithm such as Random Forests or Gradient Boosting, selecting appropriate features from these internal representations should be possible. Works as those by Minixhofer et al. (2021), Gombert and Bartsch (2021), Rotaru (2021), Smolenska et al. (2021) or Gombert (2021) show that the integration of task-specific transformer-based contextualized embeddings with more traditional feature-based algorithms can yield fruitful outcomes. Considering systems such as the ones published by Ha et al. (2019), one could easily replace the generalized embeddings they use with task-specific ones. Future work could thus involve testing whether such embeddings can add to a more traditional feature set to improve the overall predictive power of a given model.

It is also visible that the prediction of the *Item Difficulty* remains a challenging task since even the best participating models barely outperformed a dummy baseline model. On average, the models underestimate the difficulty of input items. In the case of this shared task, this effect might result from the data set being comparably small and from a highly specialized domain, namely the biomedical one with its comparably complicated and specialized language.

However, since all past work on predicting the difficulty and required response time of multiple choice questions using machine learning models was aimed at assessments from this domain, it is hard to make generalized judgements on the overall difficulty of this problem. What is required here is the publication of additional datasets from different domains and the evaluation of models using these. In this context, cross-domain evaluations especially would be of high use.

Predicting the *Item Response Time* was a more fruitful endeavour, with models outperforming the dummy baseline by a larger margin. However, with an RMSE rate of 23.92 for the best-performing model, one still needs to consider that the predicted *Item Response Time* is far from accurate. The same issue for predicting the *Item Difficulty* holds true for the *Item Response Time*: the dataset at hand is from a highly specialized domain, and data from other domains is not generally available.

5 Conclusion

This paper explains our submissions for the BEA 2024 shared task on predicting the *Item Difficulty* and the *Item Response Time*, of which the best placed first for predicting the *Item Difficulty* and fifth for predicting the *Item Response Time*. Our architecture combines pre-trained transformer encoder models with multi-task learning and custom regression heads, expanding upon an architecture published by Gombert et al. (2022) by combining them with *Scalar Mixing* and *Rational Activations*.

The results suggest predicting *Item Response Time* and especially *Item Difficulty* are comparably difficult tasks. However, the dataset used for this paper stems from the biomedical domain. This domain uses a very specialized language. For this reason, the tasks need to be evaluated with data from more domains to make a general claim. This could be the objective of future work.

6 Limitations

The limitations of our systems have already been discussed in the Discussion section. First, the dataset used is from a narrow domain. For this reason, results might not translate to datasets from other domains. So far, datasets from domains other than the medical one are unavailable. This is a clear research gap that must be addressed in future work. Second, even though our systems won one of the two shared tracks and generally achieved high ranks, the results suggest that the problems of predicting *Item Difficulty* and *Item Response Time* are far from solved.

References

Xinming An and Yiu-Fai Yung. 2014. Item response theory: What it is and how you can use the irt procedure to apply it. *SAS Institute Inc. SAS364-2014*, 10(4):1–14.

American Educational Research Association, American Psychological Association, National Council on Measurement in Education, et al. 1985. Standards for educational and psychological testing. *APA*.

Peter Baldwin, Victoria Yaneva, Janet Mee, Brian E Clauser, and Le An Ha. 2021. Using natural language processing to predict item response times and improve test construction. *Journal of Educational Measurement*, 58(1):4–30.

Lisa Beinborn, Torsten Zesch, and Iryna Gurevych. 2015. Candidate evaluation strategies for improved difficulty prediction of language tests. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–11, Denver, Colorado. Association for Computational Linguistics.

Claude Brezinski, Ufr Ieea, and Jim Van Iseghem. 1995. A taste of padé approximation. *Acta Numerica*, 4:53–103.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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.

William H DuBay. 2007. *Smart language*. Booksurge Publishing.

Sebastian Gombert. 2021. Twin BERT contextualized sentence embedding space learning and gradient-boosted decision tree ensembles for scene segmentation in german literature. In *Proceedings of the*

Shared Task on Scene Segmentation co-located with the 17th Conference on Natural Language Processing (KONVENS 2021), Düsseldorf, Germany, September 6th, 2021, volume 3001 of *CEUR Workshop Proceedings*, pages 42–48. CEUR-WS.org.

Sebastian Gombert and Sabine Bartsch. 2021. TUDA-CCL at SemEval-2021 task 1: Using gradient-boosted regression tree ensembles trained on a heterogeneous feature set for predicting lexical complexity. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 130–137, Online. Association for Computational Linguistics.

Sebastian Gombert, Daniele Di Mitri, Onur Karademir, Marcus Kubsch, Hannah Kolbe, Simon Tautz, Adrian Grimm, Isabell Bohm, Knut Neumann, and Hendrik Drachsler. 2022. Coding energy knowledge in constructed responses with explainable nlp models. *Journal of Computer Assisted Learning*, 39(3):767–786.

Le An Ha, Victoria Yaneva, Peter Baldwin, and Janet Mee. 2019. Predicting the difficulty of multiple choice questions in a high-stakes medical exam. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 11–20, Florence, Italy. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

Joseph Marvin Imperial and Ekaterina Kochmar. 2023. Automatic readability assessment for closely related languages. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5371–5386, Toronto, Canada. Association for Computational Linguistics.

Ilia Kuznetsov and Iryna Gurevych. 2020. A matter of framing: The impact of linguistic formalism on probing results. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 171–182, Online. Association for Computational Linguistics.

George F Madaus and Peter W Airasian. 1970. Placement, formative, diagnostic, and summative evaluation of classroom learning. In *Proceedings of the AERA Annual Meeting, 1970*. ERIC.

Benjamin Minixhofer, Milan Gritta, and Ignacio Iacobacci. 2021. Enhancing transformers with gradient boosted decision trees for NLI fine-tuning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 303–313, Online. Association for Computational Linguistics.

Alejandro Molina, Patrick Schramowski, and Kristian Kersting. 2020. Padé activation units: End-to-end learning of flexible activation functions in deep networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Ulrike Padó. 2017. Question difficulty – how to estimate without norming, how to use for automated grading. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–10, Copenhagen, Denmark. Association for Computational Linguistics.

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 Edouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830.

Yifan Peng, Qingyu Chen, and Zhiyong Lu. 2020. An empirical study of multi-task learning on BERT for biomedical text mining. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, pages 205–214, Online. Association for Computational Linguistics.

Georg Rasch. 1960. *Probabilistic models for some intelligence and attainment tests.*

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Armand Rotaru. 2021. ANDI at SemEval-2021 task 1: Predicting complexity in context using distributional models, behavioural norms, and lexical resources. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 655–660, Online. Association for Computational Linguistics.

Burr Settles, Geoffrey T. LaFlair, and Masato Hagiwara. 2020. Machine learning–driven language assessment. *Transactions of the Association for Computational Linguistics*, 8:247–263.

Greta Smolenska, Peter Kolb, Sinan Tang, Mironas Bitinis, Héctor Hernández, and Elin Asklöv. 2021. CLULEX at SemEval-2021 task 1: A simple system goes a long way. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 632–639, Online. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Robert Tinn, Hao Cheng, Yu Gu, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2023. Fine-tuning large neural language models for biomedical natural language processing. *Patterns*, 4(4):100729.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3261–3275.

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.

Victoria Yaneva, Peter Baldwin, Christopher Runyon, et al. 2023. Extracting linguistic signal from item text and its application to modeling item characteristics. In *Advancing Natural Language Processing in Educational Assessment*, pages 167–182. Routledge.

Victoria Yaneva, Le An Ha, Peter Baldwin, and Janet Mee. 2020. Predicting item survival for multiple choice questions in a high-stakes medical exam. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6812–6818, Marseille, France. European Language Resources Association.

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.

Victoria Yaneva, Kai North, Peter Baldwin, An Ha Le, Saed Rezayi, Yiyun Zhou, Sagnik Ray Choudhury, Polina Harik, and Brian Clauser. 2024. Findings from the First Shared Task on Automated Prediction of Difficulty and Response Time for Multiple Choice Questions. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

UnibucLLM: Harnessing LLMs for Automated Prediction of Item Difficulty and Response Time for Multiple-Choice Questions

Ana-Cristina Rogoz and **Radu Tudor Ionescu**
Department of Computer Science
University of Bucharest
14 Academiei, Bucharest, Romania
raducu.ionescu@gmail.com

Abstract

This work explores a novel data augmentation method based on Large Language Models (LLMs) for predicting item difficulty and response time of retired USMLE Multiple-Choice Questions (MCQs) in the BEA 2024 Shared Task. Our approach is based on augmenting the dataset with answers from zero-shot LLMs (Falcon, Meditron, Mistral) and employing transformer-based models based on six alternative feature combinations. The results suggest that predicting the difficulty of questions is more challenging. Notably, our top performing methods consistently include the question text, and benefit from the variability of LLM answers, highlighting the potential of LLMs for improving automated assessment in medical licensing exams. We make our code available at: https://github.com/ana-rogoz/BEA-2024.

1 Introduction

High-stakes medical licensing exams, like the United States Medical Licensing Examination (USMLE), require well-crafted questions to accurately assess a candidate's knowledge and skills. Traditionally, determining item difficulty and response time (average time to answer) relied on pretesting, which can be carried out by embedding new items alongside scored items in live exams. However, this method has been recognized as impractical due to resource limitations (Settles et al., 2020).

This year's Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024) directly addresses this problem and its resource limitations by proposing a shared task on Automated Prediction of Item Difficulty and Item Response Time for USMLE exam items. This initiative fosters collaboration and innovation in developing reliable prediction methods, while also contributing to creating more efficient, secure, and informative medical licensing exams.

This paper details our participation in the shared task (Yaneva et al., 2024), where we investigated the use of Large Language Models (LLMs) to predict difficulty and response time for retired USMLE Multiple-Choice Questions (MCQs). Our main contribution is to augment the dataset by incorporating answer choices generated by several zero-shot LLMs (Falcon, Meditron, Mistral). To solve the two prediction tasks (question response time prediction, question difficulty prediction), we employ transformer-based models that alternatively employ six different feature combinations. Our findings indicate that predicting question difficulty proves to be a more complex task. Interestingly, the most successful models consistently incorporate the question text, and benefit from the augmentation based on LLM-generated answers. Our results highlight the potential of LLMs to enhance automated assessment methods in medical licensing exams.

We also present post-competition methods that obtain better results than the originally submitted models. These newer models are aimed at addressing overfitting and our wrong choice of features.

2 Related work

The need for alternatives to the traditional processes motivates the exploration of new methods for estimating item difficulty and response time. Recent research (Ha et al., 2019; Yaneva et al., 2020; Xue et al., 2020; Baldwin et al., 2021; Yaneva et al., 2021) suggests promising results using machine learning models trained on item text data to predict these characteristics.

One of the seminal studies in this direction (Ha et al., 2019) investigated the feasibility of using machine learning models to predict both item difficulty and response time for multiple-choice questions in a high-stakes medical exam. The authors focused on extracting various features from the

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 493–502
June 20, 2024 ©2024 Association for Computational Linguistics

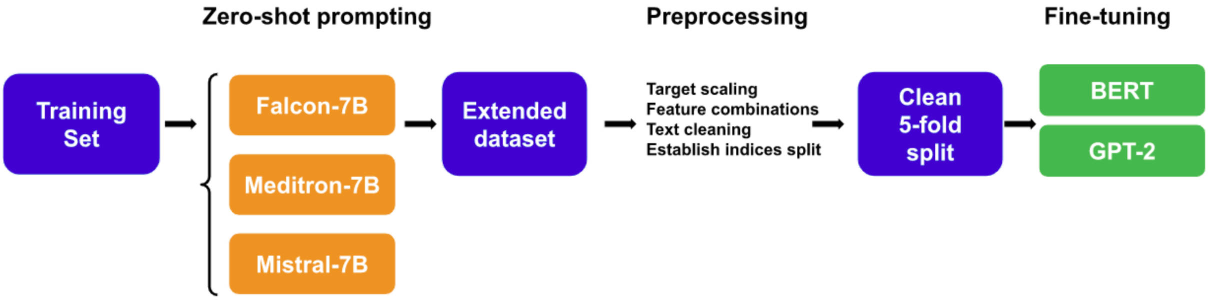


Figure 1: An overview of the data preprocessing and model training workflow for predicting item difficulty and response time of medical exam questions. The initial dataset is enriched with zero-shot prompted responses generated by Large Language Models (LLMs). We then perform preprocessing over the augmented dataset by scaling the target labels, adding new feature combinations, text cleaning and establishing the split for cross-validation. Finally, two alternative transformer-based models are fine-tuned on the augmented data.

question text data, including linguistic features and embedding types. Their models were then trained to predict these difficulty and response time characteristics. The encouraging results from this study suggest that machine learning offers a promising alternative to traditional, resource-intensive pretesting methods for estimating these important exam design elements. While the prior studies focused on predicting difficulty and response time separately, Xue et al. (2020) explored a method that could predict both simultaneously, using transfer learning. Their research suggests that this approach offers potential benefits in terms of efficiency.

In addition to predicting difficulty and response time, researchers explored another valuable application of machine learning: item survival prediction (Yaneva et al., 2020). This task focuses on estimating the likelihood of an item to be included in the final exam based on its difficulty and other question characteristics, and highlights the versatility of machine learning for various stages of exam design. Another approach was presented by Baldwin et al. (2021), who study the use of linguistic features to predict the response process complexity, which refers to the mental steps required to answer a medical MCQ.

Instead of predicting difficulty, Yaneva et al. (2021) leveraged the use of linguistic features to predict the response process complexity associated with answering medical MCQs. Their work sheds light on the underlying factors that contribute to the difficulty of these questions.

In summary, automated approaches offer several advantages, such as efficiency (predicting from text eliminates the need for pretesting, saving time and resources), security (reduced reliance on pretesting minimizes the risk of question exposure), and scalability (automated methods allow for creating larger pools of high-quality questions). Therefore, continuously validating the use of machine learning to replace traditional methods is currently an active research topic.

3 Methods

We start by annotating the original dataset with answers obtained by prompting LLMs in a zero-shot setup. The extended dataset is further processed by scaling the target labels, creating additional feature combinations, text cleaning, and setting the data split for cross-validation. The cleaned dataset is employed to fine-tune two transformer-based models. The end-to-end overview of the employed framework is presented in Figure 1. Below, we describe each step of our pipeline in detail.

3.1 Zero-Shot Prompting

We conjecture that LLMs can be employed to provide answers to the questions that need to be evaluated in terms of difficulty and response time, and the returned answers can be harnessed to better solve the prediction tasks. For instance, the number of LLMs that give correct answers to a question can be a strong indicator for the difficulty level of the respective question. To this end, we rely on three LLMs to obtain the answers, namely Falcon-7B (Almazrouei et al., 2023), Meditron-7B (Chen et al., 2023) and Mistral-7B (Jiang et al., 2023). We resort to the use of models with 7B parameters, due to our computing resource limitations. However, we compensate for the use of lighter LLMs by integrating multiple models. While Meditron-7B is specialized on the medical domain, which per-

#Item	Falcon	Meditron	Mistral
391	The correct answer is: C. Weight loss program. The correct answer is: C.	The correct answer is option A. The patient has a history of hy	The correct answer is D. Antihypertensive therapy. The patient has
148	The answer is: A. Common fibular (peroneal), The common	The correct answer is option A. The common fibular nerve is	The correct answer is A. Common fibular (peroneal). The common
562	The answer is: A. A, B. B, C. C, D.	The correct answer is option D. The correct answer is option D.	The correct answer is D. D. The patient has a

Table 1: Examples of Falcon, Meditron and Mistral answers, when prompted with USMLE questions together with the multiple-choice answers. The examples are not truncated (although it often seems so).

Feature name	Description
ItemNum	Index
ItemText	Question text
Answer_[A-J]	Multiple choice answers
Answer_Key	Single value between A-J
Answer_Text	The text of the correct answer
ItemType	Text or PIX (i.e. image)
EXAM	Step_[1, 2, 3]
Difficulty	Real value indicating question difficulty.
Response_Time	Integer value indicating mean response time (s).

Table 2: Initial set of features from the original shared-task dataset.

fectly suits the provided shared task data, the other LLMs are general purpose models. These choices are aimed at enhancing the *diversity* of the models, which was previously reported as a relevant aspect when constructing ensembles (Georgescu et al., 2023). Hence, by combining the outputs of the three LLMs, we aim to leverage the complementary strengths of all models. The selected LLMs are trained on distinct datasets, and they exhibit different capabilities in reasoning, factual recall, or creative text generation. By employing diverse models, we aim to reduce the influence of biases learned by individual models, thus achieving a higher generalization capacity. For each sample, we prompt the three LLMs in the following manner:

```
PROMPT: "You are a student taking the
USMLE exam. Your task is to answer the
following question with one of the
multiple choices.

$ItemStem_Text
A.$Answer_A,
B.$Answer_B,
..."
```

Building on the provided prompts, Table 1 showcases example responses from the three LLMs (Falcon-7B, Meditron-7B, and Mistral-7B). Interestingly, we observe a wide spectrum of agreement, ranging from all three models providing identical answers to complete divergence in their responses.

3.2 Preprocessing

To ensure consistent scaling across labels, we normalize the "Response_Time" and "Difficulty" target labels to a common range between 0 and 1. Following the scaling of target variables, we apply additional preprocessing steps to the LLM outputs. To improve performance and data consistency, we cleaned the LLM answer texts by removing any extra spaces and new line characters.

To ensure the reproducibility of our results, we provide the preprocessed and augmented dataset, containing both training and test sets, at https://github.com/ana-rogoz/BEA-2024.

3.3 Data Engineering

A detailed breakdown of the available features can be found in Table 2. We checked how well input features correlate to the target Response_Time and Difficulty values, and concluded that the EXAM, AnswerKey and ItemType columns display no correlation, as shown in Figures 2, 3 and 4, respectively. Thus, before the competition, we decided to exclude these columns from all our experiments, except for one baseline that includes all original features. However, this decision overlooks an important insight: although the AnswerKey alone does not correlate with the labels, it could represent a

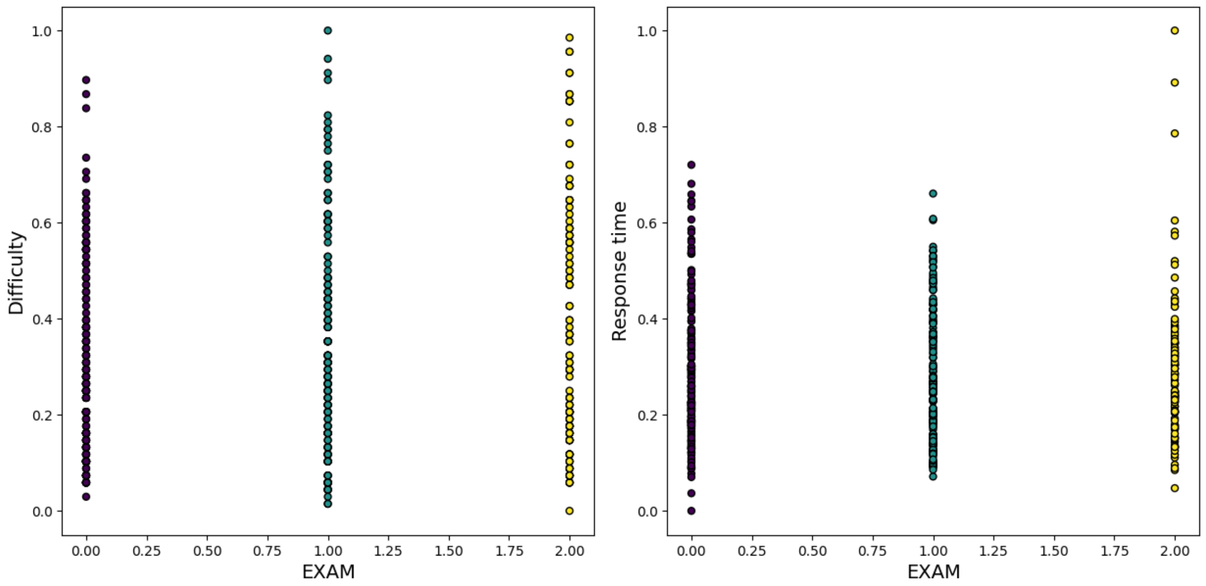


Figure 2: **Left**: Correlation between the EXAM integer feature and the difficulty label. **Right**: Correlation between the EXAM integer feature and the response time label.

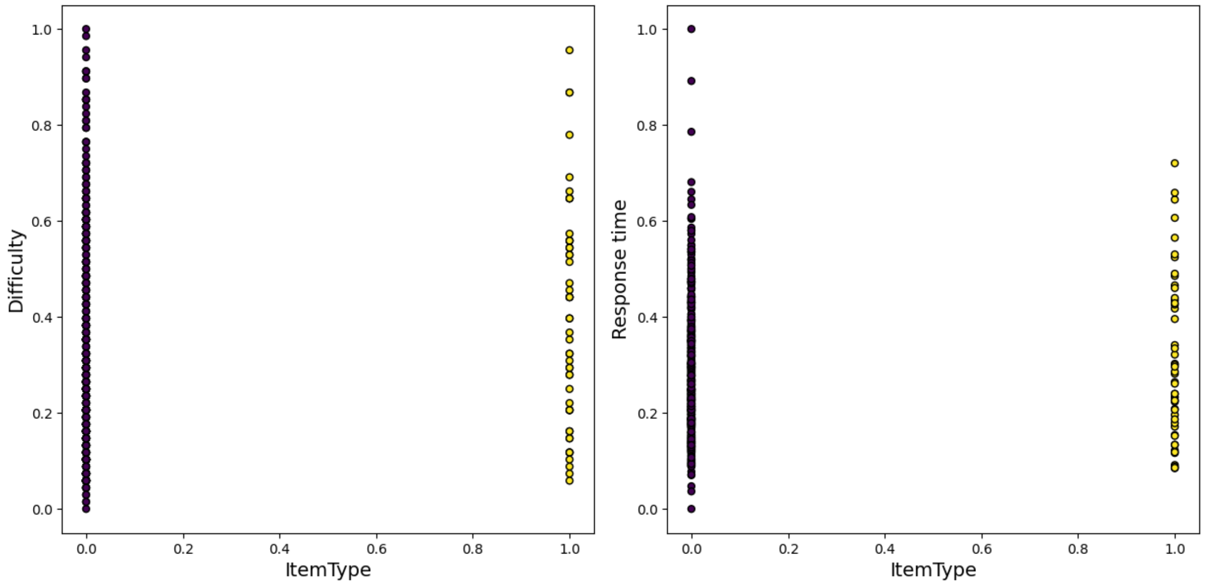


Figure 3: **Left**: Correlation between the ItemType integer feature and the difficulty label. **Right**: Correlation between the ItemType integer feature and the response time label.

very useful feature when combined with LLM answers. This is because an LLM can answer "The correct answer is D.", and comparing this answer with the AnswerKey feature can tell us if the LLM was able to correctly identify the correct answer or not. To this end, we combine the AnswerKey feature with LLM features in our post-competition models.

To enrich the input data provided to our models, we engineer seven new features, as presented in Table 3. These features combine original dataset features with the LLM-generated answers. This process aims to capture a more comprehensive representation of the problem for the trained models.

To mitigate the limitations of the very small dataset size, we employ a 5-fold cross-validation procedure to train our models. This technique involves shuffling the data and splitting it into five fixed equally-sized subsets. Each fold is then used for training and validation in turn, providing a robust evaluation of the models. The final extended and shuffled dataset is part of our publicly available

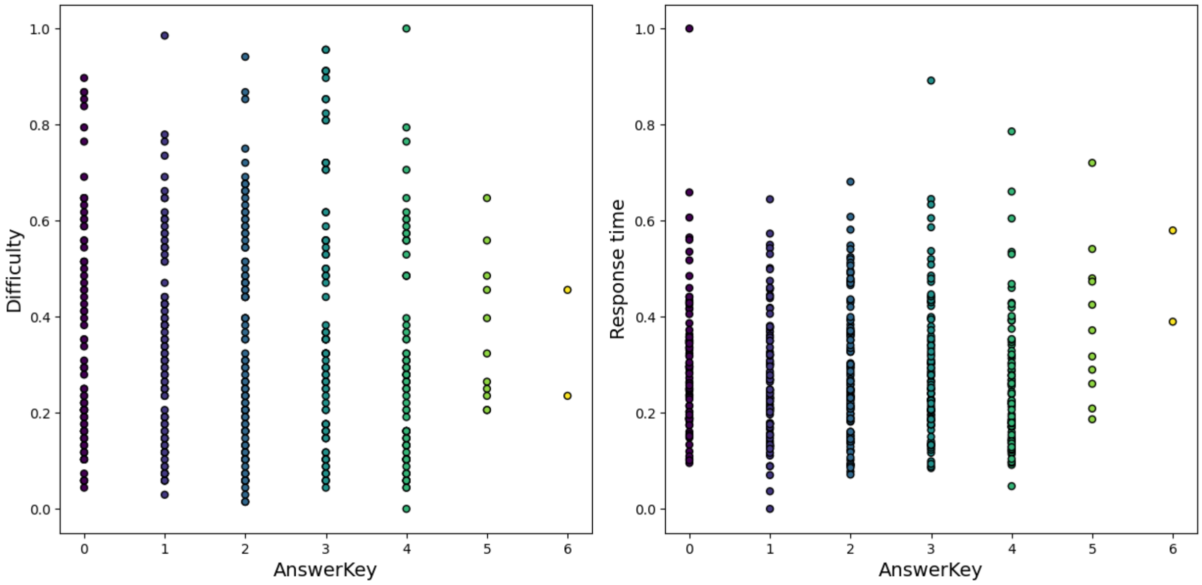


Figure 4: **Left**: Correlation between the AnswerKey integer feature and the difficulty label. **Right**: Correlation between the AnswerKey integer feature and the response time label.

Feature set	Merged features
all	All initial feature columns
q_answers	ItemText, Answer_*, Answer_Text
answers	Answer_*
q_a	ItemText, AnswerText
llms_a	LLM answers, AnswerText
q_llms_a	ItemText, LLM answers, AnswerText
q_llms_a_key$^\diamond$	ItemText, LLM answers, AnswerText, AnswerKey

Table 3: Combinations of features that are alternatively used to train our models. The $\diamond$ symbol indicates the feature set is added post-competition.

repository.

3.4 Models

Our work focuses on training and applying different models to the two regression tasks, namely predicting response time and difficulty of medical questions. We utilize two transformer-based approaches, well-suited for learning complex relationships. The models are trained on the new sets of constructed features, which are detailed in Table 3. We also include a basic linear modeling approach as baseline. After the competition, we decided to employ a model that uses frozen transformer-based features and trains only a linear model on top of the deep features. This decision is aimed at addressing the potential of overfitting transformer-based models to the very small dataset available for the competition.

Fine-tuned BERT. Our first method employs a fine-tuned Bidirectional Encoder Representations from Transformers (BERT) model (Devlin et al., 2019) for the regression tasks. We leverage the pre-trained BERT encoder to generate contextualized representations for each text input as 768-dimensional vectors. However, instead of the standard classification head, we implement a single-neuron regression head. Finally, a sigmoid activation function is applied to the output layer, ensuring the predictions fall within the desired range of 0 to 1.

Fine-tuned GPT-2. Similar to the BERT-based approach, our second method fine-tunes a GPT-2 model (Radford et al., 2019) for regression. We utilize the corresponding GPT-2 tokenizer to convert text inputs into numerical representations. The pre-trained GPT-2 model undergoes further training (fine-tuning) with a single-neuron output layer at the end. Once again, we employ a sigmoid activation function to ensure the model's predictions fall within the interval $[0, 1]$.

ν-Support Vector Regression + TF-IDF. In addition to transformer-based approaches, we investigate a linear regression method, namely ν-Support Vector Regression (ν-SVR) (Schölkopf et al., 2000). We experiment with two shallow feature extraction techniques, namely TF-IDF and

TF-IDF combined with Principal Component Analysis (PCA), focusing on the statistical properties of words in a document.

ν-Support Vector Regression + BERT. Fine-tuning large models, e.g. BERT or GPT-2, on small datasets is prone to overfitting. To mitigate overfitting, an alternative to end-to-end fine-tuning is keeping the pre-trained layers frozen, and training only the last regression layer. To this end, we propose a model that employs BERT-based embeddings and trains a ν-SVR model on top, an approach that is also known as *linear probing*. As input to the BERT model, we consider LLM answers with and without the AnswerKey feature. The resulting ν-SVR+BERT models are added post-competition.

4 Experiments

4.1 Dataset

In the BEA 2024 Shared Task, the dataset provided by the organizers consists of retired Multiple-Choice Questions (MCQs) from the United States Medical Licensing Examination. The data is divided into two distinct subsets: an initial training set of 466 samples and a separate test set of 201 samples, which is used to evaluate the participants.

4.2 Evaluation

We assess the performance levels of our methods using two complementary metrics: the mean squared error (MSE) and the Kendall τ correlation. MSE measures the average squared difference between predicted and actual values, indicating how well the model fits the data, while the Kendall τ correlation evaluates the model's ability to capture the general trend of the data, providing insights into its generalization capability.

4.3 Hyperparameter Tuning

The hyperparameters of all models are determined via grid search. For the transformer-based methods (BERT, GPT-2), we employ a grid search over the maximum number of input tokens in $\{100, 150, 200, 250, 300, 350, 400, 512\}$, learning rate values in $\{10^{-4}, 5 \cdot 10^{-4}, 10^{-5}, 5 \cdot 10^{-5}, 10^{-6}, 5 \cdot 10^{-6}\}$, and number of training epochs in $\{5, 10, 15, 20\}$. The models are optimized using the AdamW optimizer (Loshchilov and Hutter, 2019) on mini-batches of 32 samples. For the ν-SVR approaches, we employ a grid search over the parameter C in the set $\{0.01, 0.1, 0.5, 1, 5, 10, 50, 100\}$ and values of ν in $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. The complete hyperparameter setup for our experiments, as well as the methods themselves, are available as part of our publicly available repository: https://github.com/ana-rogoz/BEA-2024.

4.4 Cross-Validation Results

Due to the modest training dataset size (466 training samples), we employ 5-fold cross-validation to obtain robust evaluation results. We present the results based on the cross-validation procedure in Table 4. The results represent the average MSE and Kendall τ correlation values obtained across the 5 folds. Our experiments show a notable difference in task difficulty. Indeed, predicting difficulty proves to be significantly more challenging than predicting response time.

Response time. Our 5-fold cross-validation results indicate that the SVR+BERT models based on "q_llms_a_key" (0.0132) and "q_llms_a" (0.0134) features achieve the best MSE values for question response time prediction. They are followed by the fine-tuned BERT based on "q_answers" features (0.0148). In terms of the Kendall τ correlation, the top three models are the same, but their ranking is different. More precisely, the fine-tuned BERT based on "q_answers" features surpasses the SVR+BERT models in terms of Kendall τ.

We notice that the ν-SVR models based on TF-IDF representations struggle to learn effective relationships between features and target labels. However, this is clearly an issue of the shallow TF-IDF features, since the ν-SVR models based on BERT embeddings are at the opposite end of the performance spectrum.

Our experiments reveal two interesting findings regarding feature selection for response time prediction. First, transformer models that rely only on the multiple-choice answers obtain sub-optimal results compared with those that include the original question. This suggests that the question itself provides valuable information about the response time. The second important observation is that the AnswerKey feature becomes useful when combined with LLM answers, boosting the performance of SVR+BERT when using "q_llms_a_key" features instead of 'q_llms_a" features, with respect to both MSE and Kendall τ measures.

Difficulty. Similar to the response time prediction task, we analyze the top models for question difficulty prediction in terms of both MSE and

Task	Model	Features	MSE ↓	Kendall τ ↑	Run
Response Time	BERT	all	0.0151 ± 0.0016	0.3810 ± 0.0543	
		q_answers	0.0148 ± 0.0011	**0.4232** ± 0.0350	1
		answers	0.0190 ± 0.0017	0.1334 ± 0.0241	
		q_a	0.0149 ± 0.0010	0.3718 ± 0.0452	
		llms_a	0.0171 ± 0.0003	0.2401 ± 0.0467	
		q_llms_a	0.0150 ± 0.0012	0.3912 ± 0.0414	2
	GPT-2	all	0.0245 ± 0.0085	0.3550 ± 0.0876	
		q_answers	0.0157 ± 0.0023	0.4029 ± 0.0458	3
		answers	0.0231 ± 0.0041	0.0703 ± 0.0404	
		q_a	0.0238 ± 0.0049	0.2949 ± 0.0766	
		llms_a	0.0292 ± 0.0102	0.1417 ± 0.0497	
		q_llms_a	0.0249 ± 0.0044	0.2536 ± 0.0984	
	SVR	q_llms_a + BERT	0.0134 ± 0.0011	0.4127 ± 0.0362	∗
		q_llms_a_key + BERT	**0.0132** ± 0.0012	0.4141 ± 0.0289	∗
		q_a + TF-IDF	0.0254 ± 0.0017	0.1532 ± 0.0241	
		q_a + TF-IDF + PCA	0.0294 ± 0.0017	0.1132 ± 0.0652	
Difficulty	BERT	all_input	0.0534 ± 0.0101	0.0780 ± 0.0469	
		q_answers	0.0534 ± 0.0102	0.0570 ± 0.0862	
		answers	0.0522 ± 0.0107	0.0795 ± 0.0481	
		q_a	0.0538 ± 0.0092	0.0812 ± 0.0189	
		llms_a	0.0562 ± 0.0105	0.0204 ± 0.0610	
		q_llms_a	**0.0500** ± 0.0093	0.1470 ± 0.0447	1
	GPT-2	all_input	0.0700 ± 0.0080	0.0727 ± 0.0640	
		q_answers	0.0659 ± 0.0052	0.1155 ± 0.0208	2
		answers	0.0571 ± 0.0130	0.0323 ± 0.0518	
		q_a	0.0623 ± 0.0059	0.0802 ± 0.0507	
		llms_a	0.0707 ± 0.0377	0.1129 ± 0.0472	
		q_llms_a	0.0599 ± 0.0142	0.1259 ± 0.0333	3
	SVR	q_llms_a + BERT	0.0576 ± 0.0087	0.1102 ± 0.0665	∗
		q_llms_a_key + BERT	0.0534 ± 0.0067	**0.1592** ± 0.0616	∗
		q_a + TF-IDF	0.0551 ± 0.0033	−0.0895 ± 0.0305	
		q_a + TF-IDF + PCA	0.0614 ± 0.0025	−0.0896 ± 0.0350	

Table 4: Results based on the 5-fold cross-validation procedure of the proposed methods for the response time and difficulty prediction tasks. To select the runs for each task, we employ the Kendall τ correlation. For each task, we highlight the top three Kendall τ correlations in **red (bold)**, green, blue, respectively. We highlight the best MSE for each task in bold. The ↓ and ↑ symbols indicate when lower or upper values are better, respectively. The ∗ symbol indicates the results are added post-competition.

Kendall τ correlation. Interestingly, the models achieving the best MSE scores, namely the fine-tuned BERT models based on “q_llms_a” (0.0500) and “answers” (0.0522) features, incorporate the correct answer information. However, in terms of Kendall τ, the top models are slightly different. While the SVR+BERT with “q_llms_a_key” features (0.1592) reaches the highest correlation, the second and third best models employ “q_llms_a” features in combination with BERT (0.1470) and GPT-2 (0.1259). Notably, all these models benefit from the inclusion of questions and LLM answers. Moreover, all but one of the top models for both metrics include the question text as input. This reinforces the importance of the question itself for predicting difficulty. Furthermore, the best Kendall τ scores are obtained by models that always incorporate both the question and LLM answers. This highlights the potential of LLMs in capturing nuances beyond the provided question and answer choices, leading to more accurate predictions.

Similar to the previous task, the ν-SVR models

Task	Model	Features	RMSE ↓	MSE ↓	Kendall τ ↑	Run	Rank
Response Time	BERT	q_answers	26.846	0.0333	0.3579	1	11/34
	BERT	q_llms_a	26.768	0.0331	0.3482	2	10/34
	GPT-2	q_answers	26.073	0.0366	0.4767	3	7/34
	SVR	q_llms_a + BERT	25.621	0.0160	0.4472	$*$	5/35
	SVR	q_llms_a_key + BERT	25.613	0.0160	0.4399	$*$	5/35
Difficulty	BERT	q_llms_a	0.308	0.0654	0.2179	1	9/43
	GPT-2	q_answers	0.337	0.1031	0.0275	2	34/43
	GPT-2	q_llms_a	0.328	0.1502	0.0008	3	30/43
	SVR	q_llms_a + BERT	0.292	0.0638	0.0517	$*$	1/44
	SVR	q_llms_a_key + BERT	0.281	0.0582	0.1519	$*$	1/44

Table 5: Test results of our best performing methods for the response time and difficulty prediction tasks. We report the official evaluation metric (RMSE), along with our metrics (MSE and Kendall τ). The ↓ and ↑ symbols indicate when lower or upper values are better, respectively. The $*$ symbol indicates the results were added post-competition.

based on TF-IDF features seem to produce subpar results, given that their Kendall τ scores indicate negative correlations between predictions and target labels. However, the ν-SVR models based on BERT embeddings achieve comparable results with the fine-tuned transformer-based approaches, and one of the former models (based on "q_llms_a_key" features) performs even better in terms of Kendall τ than the top-three submitted models.

4.5 Final Test Results

For the test dataset, we report our two evaluation metrics, MSE and Kendall τ, on the normalized labels, as well as the official evaluation metric, i.e. the Root Mean Squared Error (RMSE), on the raw target labels. For the final evaluation on the official test set, we selected the top three models in terms of Kendall τ values. The corresponding results are presented in Table 5. In the same table, we also include our post-competition results.

Response time. All three submitted methods reach higher (worse) MSE values on the test set compared with the 5-fold cross-validation results, perhaps due to overfitting. The best MSE is achieved using the fine-tuned BERT model and the "q_llms_a" features (0.0331), surpassing the models based on "q_answers" features. The MSE-based ranking of the three runs on the test set is not the same as the one obtained via cross-validation. The ranking based on the Kendall τ correlation is also different, with the best model on the test set being the fine-tuned GPT-2 based on "q_answers" features (0.4767). This model also achieves better RMSE on the test set. However, for the other two submitted models, the RMSE metric is not correlated with Kendall τ. Compared with the other competitors, our best model ranked 7th out of 34 models.

Our post-competition results obtained by the ν-SVR+BERT models reveal consistent MSE and Kendall τ values across test and cross-validation evaluations. This suggests that keeping the pre-trained BERT frozen leads to a higher generalization capacity when the data available for fine-tuning is so small (less than 500 samples). Notably, calculating the RMSE on the held-out test set demonstrates that the SVR+BERT models outperform our officially submitted models, potentially obtaining a better rank (5th place out of 35).

Difficulty. The MSE values of our final submissions for the difficulty prediction task are higher (worse) for two out of three methods, when compared with the values reported during the 5-fold cross-validation experiments. The respective methods are the fine-tuned GPT-2 based on "q_answers" features and the fine-tuned GPT-2 based on "q_llms_a" features. The same two methods reach poor Kendall τ values, indicating almost no correlation between ground-truth and predicted labels. However, for our first run, which is represented by the fine-tuned BERT based on "q_llms_a" features, both MSE and Kendall τ values are comparable to the corresponding values reported using cross-validation (MSE: 0.0500 vs. 0.654, Kendall τ: 0.1470 vs. 2179). Our findings are in line with the official results based on RMSE, which show that the fine-tuned BERT based on "q_llms_a" features is our best run. Compared with the models submitted by other participants, our best model for the question difficulty task ranks 9th out of 43 models.

Our post-competition ν-SVR-based models yield superior performance compared to all three models submitted for the official evaluation. Remarkably, both post-competition models exhibit consistent MSE values between the cross-validation and test sets, hinting at the effective mitigation of overfitting which seems to affect our fine-tuned BERT and GPT-2 models. The configuration based on "q_llm_a_key" features achieves the lowest MSE of 0.0582, followed closely by the configuration based on "q_llms_a" features, with an MSE of 0.0638. This further confirms the utility of the AnswerKey feature in combination with LLM answers. Furthermore, considering the official RMSE metric, our post-competition models achieve impressive results. The SVR+BERT based on "q_llm_a_key" features attains the lowest RMSE of 0.281, followed by the version based on "q_llm_a" features with an RMSE of 0.292. These results would have positioned our post-competition models at the top of the leaderboard.

5 Conclusion

In this paper, we presented our approaches to the BEA 2024 Shared Task on Automated Prediction of Item Difficulty and Item Response Time of retired USMLE MCQs. Our main contribution is a task-specific data augmentation method based on adding answers to MCQs using LLMs prompted in a zero-shot setup. We carried out exhaustive experiments for both tasks, using two strong transformer-based models, in both fine-tuning and linear probing settings. We employed seven different types of feature combinations, while leveraging LLM-based answers. The empirical results showed four key findings. First, the difficulty prediction task is significantly harder than the response time prediction task. Second, we noticed that the top-performing approaches always made use of the question text. Third, LLM answers had a positive impact on performance, especially on the more difficult prediction task. Fourth, linear probing (training an SVR on frozen pre-trained features) shows a better generalization capacity than end-to-end fine-tuning, most likely due to the small training set available for the competition.

6 Limitations

To collect answers from the LLMs, we used a V100 GPT Colab runtime, with 78.2 GB Disk Space, which only allowed us to prompt the smallest versions of the three LLMs, each based on 7 billion parameters. Due to our resource limitations, we were not able to prompt larger LLMs, which could have led to better results.

The limited number of samples was an important challenge for the evaluated transformers, which are prone to overfitting on small datasets. The final results indicate that our models suffered from some level of overfitting. In future work, we aim to study several ways to avoid overfitting, such as using dropout, frozen layers, regularization terms, etc.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. 2023. The Falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

Peter Baldwin, Victoria Yaneva, Janet Mee, Brian E. Clauser, and Le An Ha. 2021. Using natural language processing to predict item response times and improve test construction. *Journal of Educational Measurement*, 58(1):4–30.

Zeming Chen, Alejandro Hernández-Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, Alexandre Sallinen, Alireza Sakhaeirad, Vinitra Swamy, Igor Krawczuk, Deniz Bayazit, Axel Marmet, Syrielle Montariol, Mary-Anne Hartley, Martin Jaggi, and Antoine Bosselut. 2023. MEDITRON-70B: Scaling Medical Pretraining for Large Language Models. *arXiv preprint arXiv:2311.16079*.

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*, pages 4171–4186.

Mariana-Iuliana Georgescu, Radu Tudor Ionescu, and Andreea Iuliana Miron. 2023. Diversity-promoting ensemble for medical image segmentation. In *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*, pages 599–606, New York, NY, USA. Association for Computing Machinery.

Le An Ha, Victoria Yaneva, Peter Baldwin, and Janet Mee. 2019. Predicting the difficulty of multiple choice questions in a high-stakes medical exam. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 11–20, Florence, Italy. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *Proceedings of the International Conference on Learning Representations*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Bernhard Schölkopf, Alex J. Smola, Robert C. Williamson, and Peter L. Bartlett. 2000. New support vector algorithms. *Neural computation*, 12(5):1207–1245.

Burr Settles, Geoffrey T. LaFlair, and Masato Hagiwara. 2020. Machine learning–driven language assessment. *Transactions of the Association for Computational Linguistics*, 8:247–263.

Kang Xue, Victoria Yaneva, Christopher Runyon, and Peter Baldwin. 2020. Predicting the difficulty and response time of multiple choice questions using transfer learning. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 193–197, Seattle, WA, USA → Online. Association for Computational Linguistics.

Victoria Yaneva, Le An Ha, Peter Baldwin, and Janet Mee. 2020. Predicting item survival for multiple choice questions in a high-stakes medical exam. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6812–6818, Marseille, France. European Language Resources Association.

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 Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 223–232, Online. Association for Computational Linguistics.

Victoria Yaneva, Kai North, Peter Baldwin, An Ha Le, Saed Rezayi, Yiyun Zhou, Sagnik Ray Choudhury, Polina Harik, and Brian Clauser. 2024. Findings from the First Shared Task on Automated Prediction of Difficulty and Response Time for Multiple Choice Questions. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

The British Council submission to the BEA 2024 shared task

Mariano Felice and **Zeynep Duran Karaoz**
British Council, UK
name.surname@britishcouncil.org

Abstract

This paper describes our submission to the item difficulty prediction track of the BEA 2024 shared task. Our submission included the output of three systems: 1) a feature-based linear regression model, 2) a RoBERTa-based model and 3) a linear regression ensemble built on the predictions of the two previous models. Our systems ranked 7th, 8th and 5th respectively, demonstrating that simple models can achieve optimal results. A closer look at the results shows that predictions are more accurate for items in the middle of the difficulty range, with no other obvious relationships between difficulty and the accuracy of predictions.

1 Introduction

The development of new items for high-stake exams is a complex process involving the need to meet many quality criteria. Among these, item difficulty is essential, as it fundamentally impacts the validity of test scores and the fairness of the test outcomes.

Item difficulty pertains to the ability of test items to differentiate among varying levels of test taker proficiency consistently across diverse populations (AlKhuzaey et al., 2021). Traditionally, the estimation of difficulty requires pre-testing the newly developed items on a representative sample of test takers (usually a few hundreds), as if they were in a regular exam, and empirically estimating various statistical characteristics based on their responses.

Test items that are answered correctly by either too many or too few test-takers fall outside pre-determined difficulty boundaries and hence are typically removed from consideration or undergo changes before being pre-tested again. This process, although effective, is labour-intensive, costly, and time-consuming, necessitating the collection and analysis of extensive data before any new item can be used in live exams. Additionally, as also noted by others (e.g., Ha et al., 2019; Settles et al., 2020), it is sometimes impractical, or not even possible, due to constraints on exam duration, the limited availability of testing opportunities and the logistic challenges associated with live testing.

To address these challenges, alternative approaches using Natural Language Processing (NLP) have been proposed to estimate this difficulty from the items' text. Predicting item difficulty has significant implications for the testing industry, not only leading to savings but also allowing the dynamic adaptation of tests to new populations.

In this paper, we describe our participation in the BEA 2024 shared task, aimed at predicting item difficulty for multiple-choice questions (MCQ) from a medical exam (Yaneva et al., 2024). We present experiments using three different approaches: 1) using a set of linguistic features from the items in traditional machine learning regression models, 2) using pre-trained language models with and without the addition of the aforementioned features, and 3) building an ensemble model from the output of the previous two.

2 Related work

Previous studies have adopted different methodologies to estimate the difficulty of items for assessment. A vast majority of these have focused on examining textual properties of items. While early studies have used readability indices as predictors (DuBay, 2004; Flesch, 1948), over time, studies have evolved to utilize a wider range of complexity-related features. These include surface lexical and syntactic features (such as word/sentence length, counts of clause types, etc. (Kintsch and Vipond, 2014; McNamara et al., 2014; Yaneva et al., 2017)), NLP-enabled features (François and Miltsakaki, 2012), and features aimed at capturing the cognitive aspects of language (Ha et al., 2019; Yaneva et al., 2021) and cohesion (McNamara et al., 2014).

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 503–511
June 20, 2024 ©2024 Association for Computational Linguistics

Other studies have attempted to model difficulty in terms of comprehensibility for humans. Mostly centred around the domain of language learning, such studies have primarily focused on applying readability metrics to language comprehension tests (Beinborn et al., 2014; Gao et al., 2018; Huang et al., 2017; Loukina et al., 2016; Pandarova et al., 2019). In such tests, reading passages are strongly associated with the subsequent comprehension questions, thereby establishing a correlation between the text's complexity and question difficulty (Huang et al., 2017; Loukina et al., 2016).

There have also been attempts to estimate difficulty from the perspective of cognitive processes and knowledge dimensions required to correctly respond to a question (Padó, 2017). Such approaches are mostly qualitative in nature and rely on heuristic methods which define difficulty according to the perceptions of learners, item writers and/or educators (AlKhuzaey et al., 2021) Item difficulty has also been estimated as part of automated item generation processes, for example by measuring the semantic similarity between an item's distractors and its prompt (Alsubait et al., 2013; Ha and Yaneva, 2018; Kurdi et al., 2020) or estimating the difficulty and discrimination parameters of items employed in e-learning tests (Benedetto et al., 2020).

In the context of MCQs, Ha et al. (2019) describe models using an extensive set of linguistic features and embeddings. The same set of linguistic features were used in a subsequent study by Yaneva et al. (2020), who obtained a strong baseline for item survival by filtering out items that were too difficult or too easy for the target test taker population. In our paper, we build upon previous research by replicating the linguistic features employed by Ha et al. (2019) and Yaneva et al. (2020) as well as fine-tuning a few transformer-based models.

3 Models

We investigated a range of different models for the task, namely traditional feature-based models, transformers and ensembles. The following sections describe these in detail.

3.1 Feature-based models

We extracted over a hundred linguistic features from the MCQs in our dataset, most of which come from previous work (i.e., Ha et al., 2019; Yaneva et al., 2020, 2021) but were re-implemented in Python, inspired by the codebase made available by the researchers. These features aim to capture several levels of linguistic information, ranging from basic lexical and syntactic attributes to others related to semantic, cognitive or readability characteristics of language. They also include features that look at the structural coherence of the text and the frequency of words. In addition to these, we incorporated several other predictors, such as the average similarity between the key and distractors as well as amongst the distractors themselves, and the number of distractors for a given item and exam type (i.e. Step 1, Step 2 and Step 3). All the features employed in our models are provided in Appendix A.

To obtain an initial benchmark for our experiments, we built our own internal baseline model using the ZeroR algorithm, which assigns the mean difficulty score of the training dataset to each instance (RMSE = 0.3150). Further to that, we conducted a correlational analysis between each feature and the item difficulty scores and added the top five best correlated features. These include counts of words not in top $4000, 5000, 3000, 2000$ and adjectives (with r ranging between 0.20 to 0.18), and indicate a trend that the presence of less common words and adjectives in an item may contribute to increased difficulty.

3.2 Transformer models

Given their proven performance in NLP tasks, we fine-tuned different pre-trained language models built on the transformer architecture (Vaswani et al., 2017). Since we framed the difficulty prediction task as a regression problem, we added a dense linear layer on top of the transformer to predict the difficulty value.

Our transformer models take the full text of the MCQ as the input, where the answer options are reformatted using two additional special tokens: [KEY] to introduce the key and [DIS] to introduce each distractor (see Figure 1). The embeddings for these new tokens were randomly initialised.

We experimented with four different pre-trained models: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), bioBERT (Lee et al., 2019) and XLNet (Yang et al., 2019). Given the evaluation metric for the BEA 2024 shared task was RMSE, we adopted the same metric as our loss function.

We also built versions of these models that incorporate the additional features described in Section 3.1. This was done by concatenating the values

A 13-month-old child is brought to the emergency department because of urticaria, swelling of the lips, and difficulty breathing immediately after eating an egg. A potential risk for hypersensitivity reaction is posed by vaccination against which of the following illnesses?
[DIS] Hepatitis
[KEY] Influenza
[DIS] Pertussis
[DIS] Poliomyelitis
[DIS] Typhoid fever

Figure 1: Example representation of an MCQ for our transformer models.

of the extracted features to the language model's pooler output, before being passed on to the linear regression layer.

3.3 Ensemble models

In an attempt to exploit the strength of our models, we also experimented with a number of ensemble methods. These included models that returned the *minimum*, *maximum* and *average* prediction from our best feature-based and transformer models as well as a linear regression stacking model.

4 Experiments

4.1 Setup

We experimented with a range of regression models and feature sets, which include: 1) the entire feature set, 2) top 5 features identified through correlational analysis and 3) several automated feature selection techniques, including select-k-best (k = 10), select-from-model (Random Forest Regressor) and recursive feature elimination (RFE) with 10 features to select. This allowed us to effectively assess the impact of feature selection on model performance and find the best settings.

All our regressors were implemented using the scikit-learn library (Pedregosa et al., 2011). The Random Forest Regressor, Decision Process Regressor and Extra Trees Regressors were trained with their default parameters. We used Linear Regression with no regularization and Lasso Regression with an alpha level of '0.1'. The SGD Regressor was set to focus on error minimization without penalty while the Gaussian Process Regressor utilized an RBF kernel by default. For Support Vector Regression (SVR), different linear and non-linear kernels were explored. SVR1 operated with a linear kernel, with an increased penalty parameter (C = 100) and a kernel coefficient (gamma = 0.1) while we set SVR2 to a linear kernel with a controlled number of iterations (max iter = 200). SVR3 was used with an RBF kernel and SVR4 with a polynomial kernel, both with default parameters.

Our transformer models were implemented in Pytorch using the *transformers* library by Hugging Face (Wolf et al., 2020). Training was done on an NVIDIA Tesla P100 GPU using the hyperparameters specified in Appendix B.

Our linear regression ensemble was trained on the predictions of our best feature-based and transformer models, using the predictions on our training and development set.

4.2 Data

The shared task dataset is comprised of 667 retired MCQs from past administrations of the United States Medical Licensing Examination (USMLE). USMLE consists of a series of exams (called 'Steps') administered by the National Board of Medical Examiners (NMBE) and the Federation of State Medical Boards, and is used for medical licensing in the United States. The items for the shared task came from Steps 1, 2 and 3 of the exam. Each item had a stem (i.e. the text describing the scenario), a key (correct answer) and a number of distractors (incorrect responses) which varied between 4 and 10. Each question was also accompanied by a couple of additional features, such as the Steps level and whether the original question included an image. The difficulty values ranged between 0.02 and 1.38, where higher values indicated greater difficulty. For further details about the dataset, we refer the reader to the shared task overview paper (Yaneva et al., 2024).

The training and test sets provided for the shared task comprised 466 and 201 items respectively. For our experiments, we further split the training data into a training and development set using an 80%-20% split, resulting in 372 and 94 instances respectively. No additional data was used to train our systems.

4.3 Results

Experiments reported in this section are based on our training-development split. Model performance was evaluated using Root Mean Squared Error (RMSE), in line with the shared task evaluation setup.

The performance of our feature-based models using different algorithms and feature selection methods is shown in Table 1. Two notable observations are the extreme RMSE values for the

Model	All features	Top 5	SelectKBest	SelectFromModel	RFE
RandomForest	0.3398	0.3409	0.3246	0.3175	0.3323
Linear Regressor	∞	0.3076	0.3041	0.3553	0.3276
SVR1	0.4242	0.3015	0.3048	0.3551	0.3164
SVR2	0.457	0.3083	0.3124	0.3656	0.322
SVR3	0.3506	0.3269	0.3184	0.3398	0.7024
SVR4	1.11	405.11	0.3222	0.5162	0.3238
LinearSVR	0.4031	0.3101	0.3094	0.3442	0.3073
SGDRegressor	0.3128	0.2928	0.3047	0.3076	0.3416
GaussianProcess	0.3654	0.5814	0.5845	0.4195	0.5845
DecisionTree	0.4822	0.404	0.4386	0.4381	0.4854
ExtraTrees	0.3524	0.3347	0.3241	0.316	0.3334
MLPRegressor	0.3862	0.2955	0.302	0.3241	0.3028
Lasso	0.315	0.315	0.315	0.315	0.315
ZeroR Baseline	0.3150				

Table 1: RMSE on the development set for our feature-based models, using different feature selection methods.

Linear Regressor when using all features (denoted by ∞), which was significantly higher than any other model, as well as for SVR4 when using either all features or just the top 5. Amongst all our models, Linear Regressor, SGD Regressor and MLP Regressor showed some of the lowest RMSEs, ranging from 0.2928 to 0.3076. While these outperformed the ZeroR baseline (RMSE = 0.3150), their results were comparable. For this reason, we selected the Linear Regressor using SelectKBest (RMSE = 0.3041) as our final model, given its simplicity and relatively lower error compared to other methods. This model uses the following 10 features derived from feature selection: 2 readability measures (FleshReadingEase, ColemanLiau), 6 cognitively-motivated features (average scores and ratios of content words that do not have a rating for imagability, familiarity and concreteness) and 2 frequency features (counts of content words not in top 3000 and 4000 words).

Building an optimal transformer-based model required finding the best performing pre-trained language model as well as additional hyper-parameter optimisation. A comparison of model performance using the training parameters in Appendix B is shown in Table 2. As the results suggest, BERT-based models perform better than XLNet, which shows the least convergence. Out of the best performing models, we chose RoBERTa for further hyper-parameter tuning, as it showed better average performance across our training and dev sets, something that we prioritised given the small size of our datasets.

Hyper-parameter optimisation involved fine-tuning our RoBERTa model using different values for dropout $(0.1, 0.3, 0.5)$, weight decay (0 vs 1×10^{-5}), learning rate (1×10^{-3}, 1×10^{-4}, 1×10^{-5}, 2×10^{-5}, 3×10^{-5}), additional features (all/none) and special tokens (enabled/disabled). However, none of those combinations were able to beat our initial model.

	RMSE		
Model	**Train**	**Dev**	**Average**
BERT	0.3411	0.3142	0.3276
bioBERT	0.3567	0.3057	0.3312
XLNet	0.4861	0.3366	0.4114
RoBERTa	0.3101	0.3121	**0.3111**

Table 2: Comparison of pre-trained models using the optimal hyper-parameters.

	RMSE		
Model	**Train**	**Dev**	**Average**
Minimum	0.3061	0.3072	0.3066
Maximum	0.3024	0.3091	0.3057
Average	0.2979	0.3056	0.3018
Linear regression	0.2944	0.3037	**0.2991**

Table 3: Performance of our ensemble models on the development set.

Finally, our best feature-based and transformer-based models were used to build different simple ensemble models that combined their predictions, as described in Section 3.3. The performance of these models is included in Table 3. Despite the small differences, results show that the linear regressor outperforms simpler combinations based on the minimum, maximum or average of predictions, so we use it as our final ensemble model.

5 Official evaluation results

Our submission to the shared task included the output of the best three models found in our experiments: 1) a feature-based linear regressor (FEAT), 2) a RoBERTa-based model (ROBERTA) and 3) a linear regression ensemble (ENSEMBLE) operating on the output of the previous two models. In all

Rank	Team name	Run	RMSE
1	EduTec	electra	0.299
2	UPN-ICC	run1	0.303
3	EduTec	roberta	0.304
4	ITEC	RandomForest	0.305
5	BC	ENSEMBLE	0.305
6	Scalar	Predictions	0.305
7	BC	FEAT	0.305
8	BC	ROBERTA	0.306
…	…	…	…
16	Baseline	DummyRegressor	0.311
…	…	…	…
43	ITEC	BERT-ClinicalQA	0.393

Table 4: Official performance evaluation of our models.

three cases, the final models used for our submission were re-trained using all the available training data, unlike for our optimisation experiments where we used only 80%.

An abbreviated version of the official results is included in Table 4. As we can see, results from different teams are very close, with an average RMSE of 0.3246 (SD = 0.0207). Our submitted systems ranked 5th (ENSEMBLE), 7th (FEAT) and 8th (ROBERTA), also showing little variation between them. However, it is interesting to see how the ensemble model ended up in the top 5, considering it operates on the output of the other two lower-ranked systems, which highlights the importance of model optimisation.

All of our systems were also able to beat the baseline (RMSE = 0.311), which only 35% of the systems did.

As all our systems directly or indirectly made use of linguistically-motivated features, we can also conclude that the explicit definition of features was crucial to achieve competitive results. This is in line with previous research, which has consistently found that traditional feature-based models tend to outperform deep learning models for regression tasks, especially when the amount of training data is very limited (Grinsztajn et al., 2022).

6 Analysis and discussion

This section looks at the performance of our best model (ENSEMBLE) in more detail. Prediction error for this model ranges from 0 to 0.8526, with a mean of 0.2494, with the majority of items having an absolute error under 0.4 (see Figure 2).

Correlation between gold standard difficulty vs predicted difficulty is 0.2024 ($p < .05$), which is considered weak (see Figure 3). In particular, we observe that prediction error decreases when the gold standard difficulty goes from 0 to roughly 0.4, then remains low between 0.4 and 0.6 and finally steadily increases from that point onwards, as seen in Figure 4. This reveals that the model is more accurate for values in the middle of the range and particularly inaccurate for very difficult items.

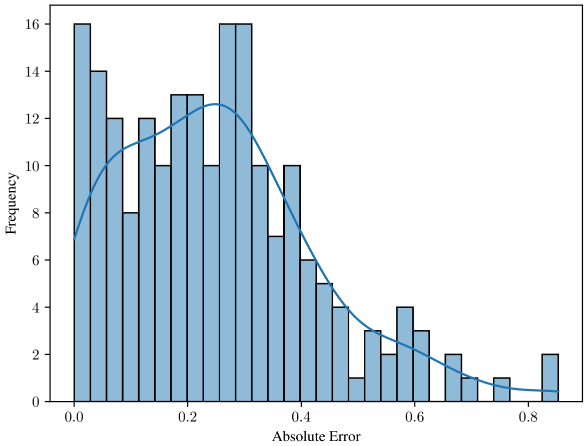


Figure 2: Distribution of prediction errors.

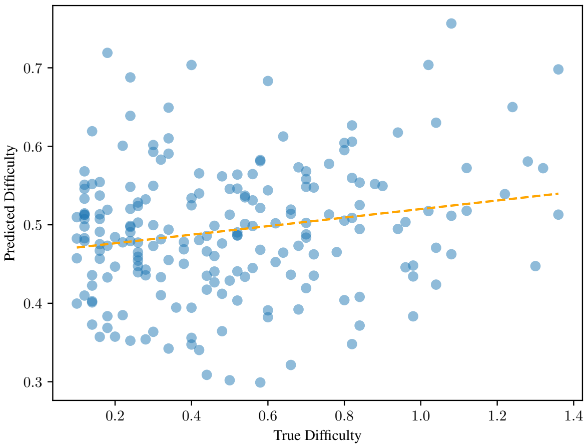


Figure 3: Correlation between gold standard difficulty and predictions by our ENSEMBLE model.

We also looked at the relationship between prediction error and item similarity, where similarity is given by the two principal components (PC1 and PC2) from Principal Component Analysis on the items' BERT embeddings (Figure 5). However, the plot shows no obvious correlations or clusters, suggesting that similar items are not predicted with the same degree of accuracy by our ENSEMBLE model.

Performance by item type shows that text-only items have a mean absolute error of 0.2506 while items with pictures yield 0.2399. Although this difference is probably negligible, it is somewhat surprising that difficulty for items containing pictures are slightly more accurately predicted when none of our models take those pictures into account

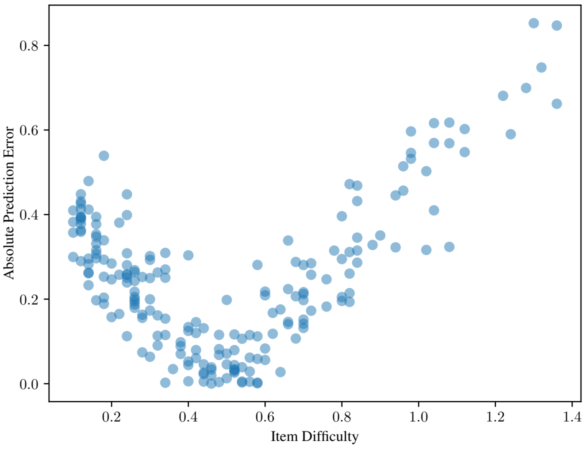


Figure 4: Gold standard difficulty vs predicted error.

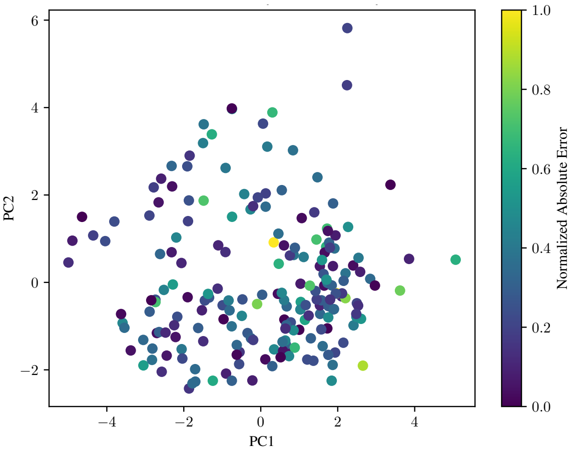


Figure 5: Prediction error and the relationship between items.

(all our models are text-based and no pictures were included in the dataset).

In terms of the exam level of each item, we found that the average prediction error increases as the Steps level is higher, which matches our intuition that that difficulty increases by level (Steps 1/2/3 mean difficulties are 0.2264, 0.2557 and 0.2782 respectively).

The effect of the number of distractors, however, does not seem to follow a clear trend, as error increases when using 4 and 7 distractors but it decreases when using 5, 6 and 8. The number of distractors yielding the lowest prediction error is 6.

7 Conclusions

In this paper, we have described the three models that were used in our submission to the BEA 2024 shared task: 1) a traditional feature-based regressor, 2) a transformer-based model and 3) an ensemble model. Our best system, a linear regressor ensemble, ranked 5th, producing near-optimal results. A detailed analysis revealed that our ensemble model is more accurate at predicting difficulty in the middle range, struggling to predict more difficult items. Other aspects, such as the inclusion of pictures or the number of distractors, do not have a significant impact on prediction accuracy.

All in all, our experiments show that simple models based on linear regression or pre-trained language models can achieve acceptable performance without excessive fine-tuning.

In future work, we would like to explore the use of custom loss functions in our transformer models as well as new features and the addition of synthetic data, since we believe that the performance of all the systems that participated in the shared task was hindered by the small size of the training data.

References

Samah AlKhuzaey, Floriana Grasso, Terry R Payne, and Valentina Tamma. 2021. A systematic review of data-driven approaches to item difficulty prediction. In *International Conference on Artificial Intelligence in Education*, pages 29–41. Springer.

Tahani Alsubait, Bijan Parsia, and Ulrike Sattler. 2013. A similarity-based theory of controlling mcq difficulty. In *2013 second international conference on e-learning and e-technologies in education (ICEEE)*, pages 283–288. IEEE.

Lisa Beinborn, Torsten Zesch, and Iryna Gurevych. 2014. Predicting the difficulty of language proficiency tests. *Transactions of the Association for Computational Linguistics*, 2:517–530.

Luca Benedetto, Andrea Cappelli, Roberto Turrin, and Paolo Cremonesi. 2020. R2de: a nlp approach to estimating irt parameters of newly generated questions. In *Proceedings of the tenth international conference on learning analytics & knowledge*, pages 412–421.

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.

William H DuBay. 2004. The principles of readability. *Online Submission*.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of applied psychology*, 32(3):221.

Thomas François and Eleni Miltsakaki. 2012. Do nlp and machine learning improve traditional readability formulas? In *Proceedings of the First Workshop on Predicting and Improving Text Readability for target reader populations*, pages 49–57.

Yifan Gao, Lidong Bing, Wang Chen, Michael R Lyu, and Irwin King. 2018. Difficulty controllable generation of reading comprehension questions. *arXiv preprint arXiv:1807.03586*.

Leo Grinsztajn, Edouard Oyallon, and Gael Varoquaux. 2022. Why do tree-based models still outperform deep learning on typical tabular data? In *Advances in Neural Information Processing Systems*, volume 35, pages 507–520. Curran Associates, Inc.

Le An Ha and Victoria Yaneva. 2018. Automatic distractor suggestion for multiple-choice tests using concept embeddings and information retrieval. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 389–398, New Orleans, Louisiana. Association for Computational Linguistics.

Le An Ha, Victoria Yaneva, Peter Baldwin, and Janet Mee. 2019. Predicting the difficulty of multiple choice questions in a high-stakes medical exam. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 11–20, Florence, Italy. Association for Computational Linguistics.

Zhenya Huang, Qi Liu, Enhong Chen, Hongke Zhao, Mingyong Gao, Si Wei, Yu Su, and Guoping Hu. 2017. Question difficulty prediction for reading problems in standard tests. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 1352–1359. AAAI Press.

Walter Kintsch and Douglas Vipond. 2014. Reading comprehension and readability in educational practice and psychological theory. In *Perspectives on memory research*, pages 329–365. Psychology Press.

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.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

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.

Anastassia Loukina, Su-Youn Yoon, Jennifer Sakano, Youhua Wei, and Kathy Sheehan. 2016. Textual complexity as a predictor of difficulty of listening items in language proficiency tests. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3245–3253.

Danielle S McNamara, Arthur C Graesser, Philip M McCarthy, and Zhiqiang Cai. 2014. *Automated evaluation of text and discourse with Coh-Metrix*. Cambridge University Press.

Ulrike Padó. 2017. Question difficulty–how to estimate without norming, how to use for automated grading. In *Proceedings of the 12th workshop on innovative use of NLP for building educational applications*, pages 1–10.

Irina Pandarova, Torben Schmidt, Johannes Hartig, Ahcène Boubekki, Roger Dale 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*, 29:342–367.

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. *Journal of machine learning research*, 12(Oct):2825–2830.

Burr Settles, Geoffrey T. LaFlair, and Masato Hagiwara. 2020. Machine learning–driven language assessment. *Transactions of the Association for Computational Linguistics*, 8:247–263.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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.

Victoria Yaneva, Le An Ha, Peter Baldwin, and Janet Mee. 2020. Predicting item survival for multiple choice questions in a high-stakes medical exam. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6812–6818, Marseille, France. European Language Resources Association.

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.

Victoria Yaneva, Kai North, Peter Baldwin, An Ha Le, Saed Rezayi, Yiyun Zhou, Sagnik Ray Choudhury, Polina Harik, and Brian Clauser. 2024. Findings from the First Shared Task on Automated Prediction of Difficulty and Response Time for Multiple Choice Questions. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Victoria Yaneva, Constantin Orăsan, Richard Evans, and Omid Rohanian. 2017. Combining multiple corpora for readability assessment for people with cognitive disabilities. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 121–132, Copenhagen, Denmark. 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.

A List of features

Group	Features
Lexical	Counts of: Words, Content Words, Content Words without Stop Words, Nouns, Verbs, Adjectives, Numbers, Commas, Complex Words (> 3 syllables), and Types (unique words); Ratios of: Content Words, Nouns, Verbs, Adjectives, Numbers, Commas, Complex Words and Types; Average Word Length in Syllables.
Readability formulae	Flesh Reading Ease, Flesh Kincaid Grade Level, Gunning Fog, Coleman Liau
Semantic	Counts of Polysemic Words; Proportion of Polysemic Words; Average Number of Senses of: Content Words, Nouns, Verbs, Adjectives and Adverbs; Average Distance to WN of Nouns, Verbs and Nouns and Verbs, Ratio of Words in WN
Syntactic	Average Length of: Sentences, Noun Phrases; Count of: Negation, Noun Phrases, Verb Phrases, Prepositional Phrases, Active Verb Phrases, Passive Verb Phrases, Agentless Passive Verbs, Relative Clauses; Ratio of: Negation, Noun Phrases, Verb Phrases, Prepositional Phrases, Passive Verbs, Active Verbs, Relative Clauses; Average Number of Words Before Main Verb, Passive Active Ratio
Cognitively motivated	Imageability, Familiarity, Age of Acquisition, Meaningfulness Ratio Colorado, Meaningfulness Ratio Paivio
Cohesion-related	Count and Ratio of: All Connectives, Temporal Connectives, Additive Connectives, Causal Connectives, Referential Pronouns
Frequency-based	Average Rank Frequency of Words and Content Words; Average Absolute Frequency of Words and Content Words; Average Relative Frequency of Words; Count of Words and Content Words Not in Top: 2000, 3000, 4000 and 5000 words
Similarity*	Path Similarity, Cosine Similarity, Levenshtein Distance, Doc Similarity, Jaccard Similarity between Stem and Key; Average Cosine and Levenshtein Similarity: Between Key and Distractors and Between Distractors
Other*	Number of Distractors, Exam Type, Item Type

Table 5: List of features employed in our study. Features marked with * have been added to those adopted from Ha et al. (2019)

B Training hyper-parameters

Learning rate	1×10^{-5}
Batch size	16
Weight decay	1×10^{-5}
Dropout	0.1
Number of epochs	3

ITEC at BEA 2024 Shared Task: Predicting Difficulty and Response Time of Medical Exam Questions with Statistical, Machine Learning, and Language Models

Anaïs Tack [1ad] **Siem Buseyne** [1bd, 2] **Changsheng Chen** [1bd]
Robbe D'hondt [1cd] [*] **Michiel De Vrindt** [1bd] [†] **Alireza Gharahighehi** [1cd]
Sameh Metwaly [1bd, 3] **Felipe Kenji Nakano** [1cd] [*] **Ann-Sophie Noreillie** [1ad]
[1] KU Leuven [a] Faculty of Arts [b] Faculty of Psychology and Educational Sciences
[c] Department of Public Health and Primary Care [d] imec research group itec
[2] Université de Lille, CIREL [3] Damanhour University

Abstract

This paper presents the results of our participation in the BEA 2024 shared task on the automated prediction of item difficulty and item response time (APIDIRT), hosted by the NBME (National Board of Medical Examiners). During this task, practice multiple-choice questions from the United States Medical Licensing Examination® (USMLE®) were shared, and research teams were tasked with devising systems capable of predicting the difficulty and average response time for new exam questions.

Our team, part of the interdisciplinary itec research group, participated in the task. We extracted linguistic features and clinical embeddings from question items and tested various modeling techniques, including statistical regression, machine learning, language models, and ensemble methods. Surprisingly, simpler models such as Lasso and random forest regression, utilizing principal component features from linguistic and clinical embeddings, outperformed more complex models. In the competition, our random forest model ranked 4th out of 43 submissions for difficulty prediction, while the Lasso model secured the 2nd position out of 34 submissions for response time prediction. Further analysis suggests that had we submitted the Lasso model for difficulty prediction, we would have achieved an even higher ranking. We also observed that predicting response time is easier than predicting difficulty, with features such as item length, type, exam step, and analytical thinking influencing response time prediction more significantly.

1 Introduction

In the medical domain, standardized tests act as crucial gatekeepers, allowing only the best healthcare professionals into the field. An example is the *United States Medical Licensing Examination*® (USMLE®), a high-stakes exam administered by the National Board of Medical Examiners (NBME) to assess a medical student's ability to provide safe and effective patient care. However, for these exams to accurately gauge the competency of medical students, organizations like the NBME meticulously design their assessments, with a specific focus on balancing the difficulty and response time of exam questions. This is essential for ensuring the fairness and validity of the exams, as test items should cover a wide range of difficulty levels, and each question should be allocated an appropriate amount of time.

Prior studies by NBME researchers have shown that predicting the difficulty and response time of medical exam questions is a challenging task (Ha et al., 2019a; Xue et al., 2020; Yaneva et al., 2020, 2021). As a result, the NBME launched an international challenge where they provided researchers with a set of retired exam questions from the USMLE®. Research teams were tasked with developing a system or model that takes as input a multiple-choice question and produces as output two estimates: (a) how challenging it is for test-takers and (b) how long it would take them to respond (see Figure 1 for an illustration). The comprehensive details and results of this shared task are outlined in the overview paper authored by Yaneva et al. (2024).

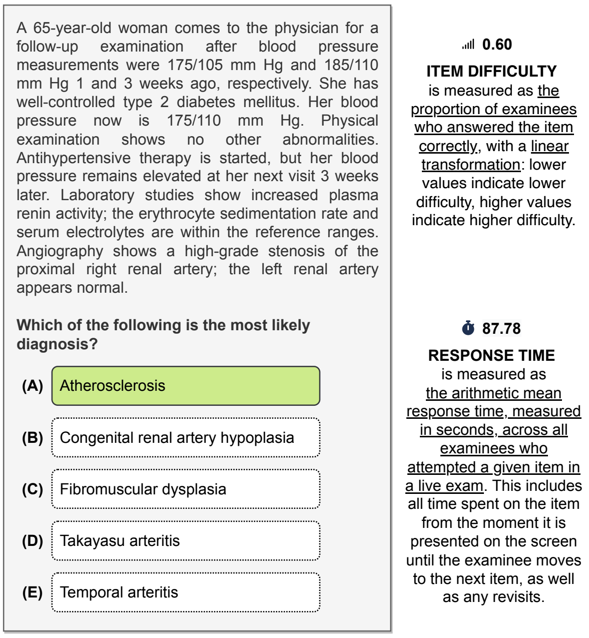
We participated in the competition with the **ITEC**[1] **team**, an interdisciplinary research group affiliated with KU Leuven and imec. Our collaborative efforts span various fields, including artificial intelligence, educational sciences, language technology, machine learning, psychometrics, and statistical modeling. Our strategy involved a fusion of statistical models, machine learning models, and language models. We integrated traditional

[*] supported by a fellowship from the Research Foundation Flanders (FWO)

[†] supported by a Baekeland mandate from the Flanders Innovation & Entrepreneurship (VLAIO)

[1] https://itec.kuleuven-kulak.be

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 512–521
June 20, 2024 ©2024 Association for Computational Linguistics



A 65-year-old woman comes to the physician for a follow-up examination after blood pressure measurements were 175/105 mm Hg and 185/110 mm Hg 1 and 3 weeks ago, respectively. She has well-controlled type 2 diabetes mellitus. Her blood pressure now is 175/110 mm Hg. Physical examination shows no other abnormalities. Antihypertensive therapy is started, but her blood pressure remains elevated at her next visit 3 weeks later. Laboratory studies show increased plasma renin activity; the erythrocyte sedimentation rate and serum electrolytes are within the reference ranges. Angiography shows a high-grade stenosis of the proximal right renal artery; the left renal artery appears normal.

Which of the following is the most likely diagnosis?

(A) Atherosclerosis

(B) Congenital renal artery hypoplasia

(C) Fibromuscular dysplasia

(D) Takayasu arteritis

(E) Temporal arteritis

0.60

ITEM DIFFICULTY
is measured as the proportion of examinees who answered the item correctly, with a linear transformation: lower values indicate lower difficulty, higher values indicate higher difficulty.

87.78

RESPONSE TIME
is measured as the arithmetic mean response time, measured in seconds, across all examinees who attempted a given item in a live exam. This includes all time spent on the item from the moment it is presented on the screen until the examinee moves to the next item, as well as any revisits.

Figure 1: Example of a multiple-choice question from the USMLE® Step 1 provided by the NBME during the shared task's training phase. Each question had an item stem and up to ten possible answers and was labeled with item difficulty and average response time.

feature engineering with contemporary fine-tuning and transfer learning approaches. The following sections will delve into our method and results.

2 Method

The shared task unfolded into two phases. During the training phase, spanning from January 15 to February 9, 2024, we received 466 multiple-choice questions along with additional metadata, such as the item type and exam step. Our goal in this phase was to develop models that could predict two key targets: item difficulty and response time. Transitioning to the evaluation phase, which took place from February 10 to February 16, 2024, we received an additional set of 201 multiple-choice question items, accompanied by the same supplementary metadata, excluding the two targets. Utilizing our top-performing models from the previous phase, our focus was on predicting the unknown targets of difficulty and response time. Each target allowed up to three final predictions to be submitted, and the submissions were then ranked based on the Root Mean Squared Error (RMSE). In this section, we provide more detailed information about our methodology.

2.1 Feature Extraction

As an initial step, we began by extracting features from the multiple-choice questions. Drawing from prior studies (e.g., Ha et al., 2019b), we employed various methods to transform the test items and answer choices into meaningful representations.

First, we used features that we could extract and compute directly from the data provided by the organizers. We defined a set of raw features including the **answer key** (A, B, C, D, E, F, G, H, I, or J), **item type** (Text or PIX), **exam** (Step 1, 2, or 3), the **number of answer options** (4, 5, 6, 7, 8, 9, or 10), the **ordinal position of the correct key within the sequence of answers options**, normalized between 0 and 1 (0.0, 0.11, 0.17, 0.2, 0.25, 0.29, 0.33, 0.4, 0.43, 0.5, 0.57, 0.6, 0.67, 0.75, 0.8, or 1.0).

Apart from the initial set of basic features, we generated more sophisticated features using various natural language processing tools. These tools encompass **Linguistic Inquiry and Word Count 2022** (LIWC-22; Pennebaker et al., 2022), evaluations of lexical sophistication relying on **TAALES 2.2** (Kyle and Crossley, 2015), and the extraction of text embeddings with the **Bio_ClinicalBERT** model (Alsentzer et al., 2019).

2.1.1 Linguistic Inquiry and Word Count

LIWC-22, created by Pennebaker et al. (2022), is a text analysis tool that facilitates the exploration of diverse linguistic dimensions within textual data. Its utility extends across various fields, including psychology and communication.

LIWC-22 offers variables including word count (total words in a text), words per sentence (average number of words per sentence), big words (percentage of words with seven letters or more), and dictionary words. The 2022 version employed in this study also evaluates newer summary variables such as analytical thinking (Pennebaker et al., 2014), clout, authenticity, and emotional tone. These metrics, derived from previous research, are calculated using standardized scores from extensive comparison corpora (Boyd et al., 2022).

In addition to the summary variables, LIWC provides valuable insights into linguistic dimensions by examining the relative frequencies of different word categories such as personal pronouns and negations, represented as percentages.

For this study, LIWC features were independently extracted for (1) the item stem of the multiple-choice question and (2) the aggregated answer options.

2.1.2 Lexical Sophistication

TAALES 2.2, developed by Kyle and Crossley (2015), is a tool designed for the automated analysis of lexical sophistication, calculating over 400 measures in this domain. Its indices have found applications in various fields such as educational psychology, cognitive science, and artificial intelligence. The tool addresses challenges associated with both second language (L2) and first language (L1) writing proficiency, L2 speaking proficiency, as well as spoken and written lexical proficiency.

The five areas of lexical sophistication covered by TAALES 2.2 include lexical frequency, range (indicating how widely a word or word family is used), n-gram frequency (measuring the frequency of combinations of n number of words), academic vocabulary, and psycholinguistic word properties (e.g., age of acquisition, concreteness, familiarity).

The tool takes a single text as input and produces a list of features for that text. In our study, we utilized the tool to extract the same set of features for five distinct input types: (1) for the item stem text, (2) for the item stem text combined with the correct answer, (3) for all the answer options combined, (4) for the correct answer, and (5) for the combined distractors.

2.1.3 Clinical Embeddings

In addition to the interpretable linguistic features outlined in Sections 2.1.1 and 2.1.2, we also considered the feature dimensions of clinical embeddings extracted from the publicly available pre-trained **Bio_ClinicalBERT** model (Alsentzer et al., 2019). These embeddings consist of 768-dimensional vectors for each token within an input text. We extracted identical features for four distinct input types:

1. For the item stem text.
2. For the scenario extracted from the item stem text (i.e., the clinical case description, excluding the final sentence; e.g., *A 65-year-old woman comes to the physician for a follow-up examination (...) the left renal artery appears normal.* in Figure 1).
3. For the question extracted from the item stem text (i.e., retaining only the final sentence in the item stem text; e.g., *Which of the following is the most likely diagnosis?* in Figure 1).
4. For each of the at most ten different answer options separately.

For each of these input types, we used Hugging Face's feature extractor pipeline to extract token embeddings and compute the average vector over all token embeddings in the input.

2.1.4 Features Summary

Utilizing the features outlined in Sections 2.1.1 to 2.1.3, we obtained a total of 4,479 features for each of the 466 multiple-choice questions in the training set. These features encompass:

1. 5 raw features
2. 235 LIWC-22 features (118 for the item stem text, 117 for the answers)
3. 1,166 TAALES 2.2 features (202 for the item stem text, 241 for the item stem text combined with the correct answer, 241 for all answer options combined, 241 for the correct answer, and 241 for the combined distractors)
4. 3,072 clinical features extracted from the BERT embeddings (768 for the scenario, 768 for the question, 768 for the correct answer, 768 for the aggregated distractors)

2.2 Model Development

Following the extraction of a comprehensive set of features, as outlined in the preceding section, our next step involved the development of various models. We conducted experiments with both statistical (see Section 2.2.1) and machine learning (see Section 2.2.2) models utilizing the set of extracted features (refer to Section 2.1). Additionally, we explored the fine-tuning of biomedical and clinical language models (see Section 2.2.3). Furthermore, we constructed an ensemble model (detailed in Section 2.2.4) by leveraging the strengths of these diverse models. Finally, we ran some feature importance analysis (Section 2.4).

2.2.1 Statistical Models

In statistical models, adhering to Occam's Razor principle (Ortner and Leitgeb, 2011), the goal was to find a simple yet effective model through two steps: filtering features and building models using the stepwise regression procedures (Venables and Ripley, 2002). The two steps were conducted on pre-processed data, where all features were normalized to maintain consistency in their scale. Additionally, features with missing values were excluded from the analyses. Ultimately, 3,952 features were utilized for the subsequent analyses conducted under 10-fold cross-validation (see Section 2.3).

Specifically, within each cross-validation fold, we initially conducted feature selection based on Pearson's correlation coefficients between the target and all features, setting a minimum threshold of 0.12 (Lovakov and Agadullina, 2021). This step aimed to identify the most relevant features, considering that stepwise regression procedures for building models could become unstable with an excessively high number of features. The number of selected features ranged from 60 to 90 across folds. Next, the selected features underwent stepwise regression analysis, which involved developing a series of simple linear regression models by iteratively removing or adding features to the baseline model. These models were then compared based on the information criteria, Akaike Information Criterion (AIC; Akaike, 1974) and Bayesian Information Criterion (BIC; Schwarz, 1978). The final selected model in each fold was determined based on the lowest value of either AIC or BIC. Since AIC or BIC could recommend different models for each fold, we calculated the average RMSE across the 10 folds to compare them. Interestingly, we found that the models recommended by BIC yielded a lower RMSE compared to those recommended by AIC. Finally, with the correlation filtering and BIC setting applied, 10 simple regression models were recommended for item difficulty and response time tasks respectively. The final chosen model was the one with the lowest RMSE across all folds.

2.2.2 Machine Learning Models

The machine learning pipeline consisted of a dimensionality reduction step followed by a model fitting step. As dimensionality reduction, we used a separate principal component analysis (PCA) for each extracted feature set (i.e., LIWC, TAALES, and BERT). The number of principal components retained for each feature set equaled the number of components required to explain at least 60% of the variance in the original features. On these preprocessed features, we trained 4 different machine learning models: Lasso (regularized linear regression), Random Forest (RF), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN).

The Lasso model was used with regularization $\alpha = 0.1$. RF was used with default hyperparameters. The hyperparameters for SVM and KNN were tuned using a grid search with nested 5-fold cross-validation. For SVM we considered an RBF kernel with regularization parameter $C \in \{0.1, 1, 10, 100\}$ and kernel width $\gamma \in \{1, 0.1, 0.01, 0.001\}$. For KNN we considered the number of neighbors $K \in \{1, 5, 10, 15, 20, 100\}$.

2.2.3 Language Models

In addition to traditional statistical and machine-learning models, we also experimented with fine-tuning a transformer model to predict response time and item difficulty as a multi-target prediction task. Previously, Xue et al. (2020) utilized a pre-trained model for a similar purpose, demonstrating the benefits of transfer learning in enhancing predictions. However, our methodology diverged in two important ways. On the one hand, we framed this as a multi-target regression task, contrary to treating response time and item difficulty as separate regression tasks, thus capturing their interdependencies. This approach is particularly meaningful as the relationship between the two variables is not strictly linear (Yaneva et al., 2021, p. 223).

On the other hand, we deliberately selected domain-specific pre-trained models tailored for biomedical or clinical texts, known to outperform nonspecific models (Alsentzer et al., 2019). The domain-specific pre-trained language models under consideration were trained on datasets from clinical sources such as MIMIC-III, as well as biomedical corpora like PubMed and PMC full-text articles and abstracts. These models encompassed **BERT-ClinicalQA** (exafluence, 2021), **Bio_ClinicalBERT** (Alsentzer et al., 2019), **Bio_ClinicalBERT_emrqa** (aaditya, 2022), **Clinical-BigBird** (Li et al., 2022), **Clinical-Longformer** (Li et al., 2023), and **ClinicalBERT** (Wang et al., 2023).

For model training, we initiated the pre-trained models sourced from Hugging Face (version 4.36), employing a PyTorch backend (version 2.1). The models were fine-tuned on an NVIDIA GeForce RTX 3090 (CUDA 12.2). We employed a BERTForSequenceClassification architecture equipped with two regression outputs, tailored for predicting both item difficulty and response time. We utilized the RMSE loss function to minimize the predicted item difficulty and response time over three epochs, assigning equal weight to both targets within the loss function. The optimization process employed the AdamW optimizer with a learning rate of 5^{e-5}, alongside a linear scheduler and weight decay. To accommodate the LongFormer and Big-Bird models, a batch size of one was used.

It should be noted that, given the substantial difference in scale between the two targets, we rescaled response time from seconds to minutes before training, thereby aiding smoother model convergence. Subsequently, during the inference stage, response time was transformed back from minutes to seconds for accurate interpretation.

It is also important to note that we chose to utilize a fixed initialization seed (15012024) for conducting post-hoc predictions after the winner announcement, aiming to ensure the reproducibility of the final reported predictions. However, it is important to acknowledge that the absence of a more comprehensive hyperparameter search on model initialization represents a limitation we intend to address in future work.

We conducted experiments using two different input formats: (1) solely focusing on the item stem and (2) concatenating the item stem with the list of answer options. Initial results suggested that including the answers led to slightly improved predictions across all models.

Moreover, we investigated whether integrating the classification of the exam step as an auxiliary task could improve the accuracy of predicting item difficulty and response time. To facilitate this classification, we introduced three extra output dimensions, indicating the probability of belonging to each exam step. This model, denoted as Bio_ClinicalBERT_FTMT, was initialized with Bio_ClinicalBERT (Alsentzer et al., 2019) and was optimized over ten epochs.

2.2.4 Ensemble Model

Motivated by the ‘no free lunch’ theorem (Wolpert and Macready, 1997), we aimed to leverage the predictive power of the diverse models introduced in the previous sections, including statistical, machine learning, and language models. The goal was to create an ensemble where individual models, each with its specific errors, could compensate for one another. Following the stacking concept, we used predictions from all individual models on training instances as features in the ensemble model.

Consistent with our approach in machine learning models, we applied dimensionality reduction to the extracted features (i.e., LIWC, TAALES, and BERT) using PCA. These reduced features were then incorporated into the ensemble model as part of its input. As for the choice of ensemble model, we experimented with Lasso, RF, Extra Trees, multi-layer perception, and gradient boosting regressor.

2.3 Model Selection

During the training phase, we ran a 10-fold cross-validation experiment on the training data, utilizing the *scikit-learn* library. The data was divided into ten folds, with these identical folds utilized for both training and evaluating each of the models outlined in Section 2.2. To maintain consistency, we utilized a fixed random seed (15012024) for shuffling the data before the splitting process. Subsequently, we calculated the average RMSE to assess and compare the performance of our various models.

2.4 Feature Importances

To better understand how the models used the input features, we performed some post-hoc interpretation techniques. One of the model-agnostic tools that we used is a permutation feature importance analysis. Such an analysis first randomly shuffles (i.e., permutes) the values for one of the features in the dataset. Then, using the models trained before on the non-shuffled data, cross-validated predictions can be regenerated with the shuffled feature and performance can be recalculated. In this way, we can see the impact on the performance of the model when one of the input features is ‘randomized’, and thus get a univariate feature importance metric. To counter variability, the whole procedure is repeated 5 times per feature with a different random permutation each time, and the average impact on performance is then reported.

3 Results

3.1 Phase 1: Cross-Validation

Regarding the difficulty, as reported in Figure 2, the Ensemble and RF methods managed to provide slightly superior results. However, the difference in performance was rather subtle, as most of the models reached RMSE values of approximately 0.30 in most of the cases. Slightly differently from that, Bio_ClinicalBERT_emrqa and Clinical-Longformer performed marginally worse by achieving an RMSE of 0.31, followed by the statistical model and Bio_ClinicalBERT_FTMT, which yielded roughly 0.32 and 0.33 of RMSE, respectively.

As for the response time, as can be seen in Figure 3, a more noticeable difference in performance was observed where Lasso and BERT-ClinicalQA

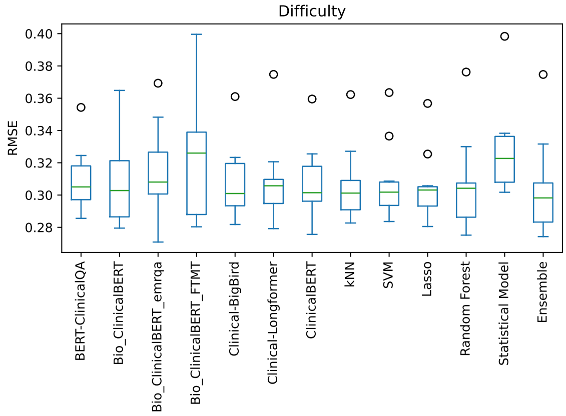


Figure 2: Performance of models in predicting difficulty on the training set, evaluated with 10-fold cross-validation.

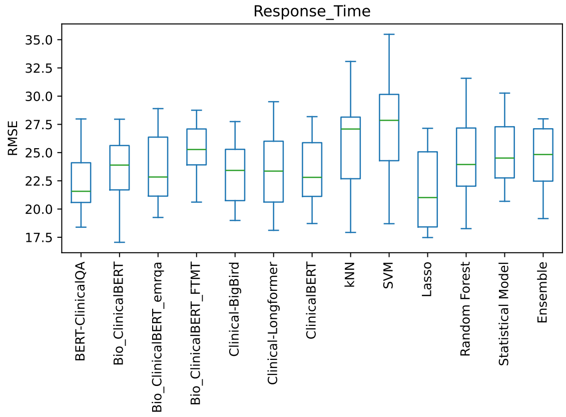


Figure 3: Performance of models in predicting response time on the training set, evaluated with 10-fold cross-validation.

had the upper hand since both of them achieved an RMSE score of approximately 21. As opposed to that, Bio_ClinicalBERT_FTM, KNN, and SVM performed relatively poorly, reaching RMSE values above 27.5. All the other compared methods provided rather overlapping results.

Submission Strategy At the end of the training phase, we devised a submission strategy. This plan entailed submitting our top two models for each track, with the final submission reserved for a model distinct from the leading two. This approach was particularly crucial for the difficulty prediction, given its inherent complexity and the difficulty in discerning superior models during the training. With all models demonstrating performance close to an average baseline (which we calculated ourselves), uncertainty arose regarding which models would outperform on the test set. Therefore, maximizing the diversity in our model selection became paramount.

For the difficulty prediction, our Ensemble and RF models achieved the lowest RMSE values. Consequently, these two models were chosen for submission during the test phase. To introduce diversity into our approach, we included BERT-ClinicalQA in the final run submission because its predictions were different from the previous runs and it was one of the top models for predicting response time. In terms of the response time prediction, our Lasso and BERT-ClinicalQA models achieved the highest scores, exhibiting the lowest RMSE values, and were thus submitted during the test phase. Additionally, to further diversify our strategy, we utilized the final run to submit a completely different model, i.e., Bio_ClinicalBERT_emrqa.

3.2 Phase 2: Leaderboard

Tables 1 and 2 present the final evaluation results for all teams based on the test datasets released by the leaderboard. For the difficulty prediction (See Table 1), the baseline model achieved an RMSE of 0.311. Our team's RF model reached 0.305, which was better than the baseline. Compared to the RMSE results of other teams, the RF model was in the 4th place out of 43 teams, and its RMSE was slightly higher than the best (0.299). Apart from our best model, our team's ensemble model also had a good performance, with an RMSE of 0.308 (slightly better than the baseline), ranking 12th out of 43 teams. For the response time prediction (See Table 2), most teams achieved better results in terms of RMSE compared to the baseline model (31.68). Our team's Lasso model performed impressively better than other models, coming the 2nd out of 34 teams with an RMSE of 24.116, significantly better than the baseline and close to the best RMSE.

Tables 3 and 4 show the performance results of all our models on the test set. It is evident from these findings that the Lasso model exhibited superior predictive capability for both difficulty and response time. Despite our knowing of the Lasso model's effectiveness during the cross-validation experiment (Figure 3) and subsequent winner announcement (Table 2), its unexpected success in predicting difficulty on the test set was surprising. Throughout the training phase, we encountered challenges in distinguishing between models for predicting difficulty, as all models performed simi-

#	Team	Run	RMSE
1	EduTec	electra	0.299
2	UPN-ICC	run1	0.303
3	EduTec	roberta	0.304
4	ITEC	RandomForest	0.305
5	BC	ENSEMBLE	0.305
12	ITEC	Ensemble	0.308
16	Baseline	DummyRegressor	0.311
43	ITEC	BERT-ClinicalQA	0.393

Table 1: Our three submissions to the leaderboard on difficulty prediction. The top 5 submissions are given as well as the shared task baseline.

#	Team	Run	RMSE
1	UNED	run2	23.927
2	ITEC	Lasso	24.116
3	UNED	run1	24.777
4	UNED	run3	25.365
5	EduTec	roberta	25.64
25	Baseline	DummyRegressor	31.68
32	ITEC	BERT-ClinicalQA	53.844
33	ITEC	Bio_ClinicalBERT_emrqa	54.719

Table 2: Our three submissions to the leaderboard on the response time prediction. The top 5 submissions are given as well as the shared task baseline.

#	Model	RMSE
	FEATURE-BASED MODELS	
4	Lasso ∗	0.301
7	Random Forest •	0.305
9	kNN	0.307
11	SVM	0.310
12	Statistical Model	0.343
	FINE-TUNED LANGUAGE MODELS	
1	Clinical-Longformer ◦∗	0.294
2	ClinicalBERT ◦∗	0.299
3	Bio_ClinicalBERT ◦∗	0.300
5	Bio_ClinicalBERT_emrqa ◦∗	0.302
6	Clinical-BigBird ◦∗	0.303
8	BERT-ClinicalQA •◦	0.306
13	Bio_ClinicalBERT_FTMT	0.350
	ENSEMBLE MODEL	
10	Ensemble •	0.308

Table 3: Performance and ranking of our models in predicting difficulty on the test set. Models denoted by • were submitted to the leaderboard. Models marked with ◦ are reported with post-hoc predictions. Models labeled with ∗ surpassed our best leaderboard model.

larly close to baseline levels. This initial difficulty hindered our recognition of the Lasso model as the optimal choice, despite its strong performance in predicting response time. Had we submitted the Lasso model to the difficulty leaderboard, we would have outperformed the second-best model, securing the second position on difficulty as well. As the Lasso model demonstrated superior performance in predicting both difficulty and response time, we will delve deeper into examining the feature importance of this model in the subsequent section.

3.3 Feature Importances

The best models based on cross-validated training RMSE turned out to be the RF for item difficulty and the Lasso for response time. Therefore, we conducted a permutation feature importance analysis for these two models. For the item difficulty, the top features for the RF were the word count from LIWC (with an average increase in RMSE of 0.034 when this feature is randomly shuffled), one of the BERT answer embeddings (0.022), one of the BERT distractor embeddings (0.017), and the analytical thinking measure from LIWC (0.016). All other features lead to an RMSE increase of at most 0.010. For the response time, the top features of the Lasso model were the word count from LIWC (with an average increase in RMSE of 6.8), the exam step (1.0), the item type (0.69), the number of answers (0.60), the analytical thinking measure from LIWC (0.30), and the position of the correct answer (0.20). All other features lead to an RMSE increase of at most 0.03.

Both these models also have built-in feature importance metrics: the RF through the heuristic values observed during training and the Lasso model through the magnitude of its coefficients. These metrics revealed that most of the total feature importance weight for the RF and Lasso models was given to the principal components (PCs) coming from the BERT embeddings (78% and 76% respectively). However, these PCs also represent 43 out of the 58 features (74%) remaining after PCA. For the RF model, each feature had a similar importance of on average 2.0% ± 1.0% (mean ± standard deviation). On the other hand, for the Lasso model, there

#	Model	RMSE
FEATURE-BASED MODELS		
1	Lasso ●	24.116
8	Random Forest	26.527
11	Statistical Model	27.020
12	kNN	28.919
13	SVM	31.101
FINE-TUNED LANGUAGE MODELS		
2	Clinical-Longformer ○	24.829
4	ClinicalBERT ○	25.643
5	BERT-ClinicalQA ●○	26.014
6	Bio_ClinicalBERT ○	26.310
7	Bio_ClinicalBERT_FTMT	26.504
9	Clinical-BigBird ○	26.555
10	Bio_ClinicalBERT_emrqa ●○	26.771
ENSEMBLE MODEL		
3	Ensemble ○	25.298

Table 4: Performance and ranking of our models in predicting response time on the test set. Models denoted by ● were submitted to the leaderboard. Models marked with ○ are reported with post-hoc predictions.

was a clear ranking of feature sets, with the raw features first (4.7% ± 4.3%) followed by the BERT PCs (1.8% ± 1.4%) and the LIWC PCs (0.10% ± 0.069%). Interestingly, while the LIWC PCs seem to have a low importance to the Lasso model based on their coefficients, they had a big impact on predictive performance based on the permutation feature importance test.

4 Discussion

As previous research by the shared task organizers has shown (Ha et al., 2019a; Xue et al., 2020; Yaneva et al., 2020, 2021), predicting response time and difficulty of multiple-choice questions for medical licensing exams is a challenging task. In this study, our team tried to solve this challenge by adopting a multidisciplinary perspective, combining insights from statistical modeling, machine learning, and natural language processing.

While previous studies have primarily concentrated on examining the influence of exam and item metadata, along with certain linguistic complexity features (e.g., Ha et al., 2019a; Yaneva et al., 2021), we explored the integration of several novel, unexplored features. While our results validate the importance of specific raw metadata features (such as the number of answer options), they also highlight the significance of features derived from LIWC and TAALES, as well as embeddings from biomedical language models. Notably, the LIWC feature indicating the degree of "analytical thinking" for answers emerged as particularly noteworthy for predicting response time.

Regarding the models, it is noteworthy that the more sophisticated ones did not surpass the less intricate models. Simple models proved more accurate in predicting the response time of multiple-choice questions. This resonates with Occam's Razor principle, which favors simpler models as long as their performance matches or exceeds that of more complex alternatives (e.g., Ortner and Leitgeb, 2011). In our study, models utilizing Lasso or RF with principal component features outperformed the fine-tuned language model with embeddings. This suggests that, for this specific task, traditional machine learning methods incorporating dimensionality reduction were more effective and robust compared to complex statistical models.

5 Conclusion

Our team's contribution to the shared task of predicting the difficulty and response time of medical exam questions demonstrates that simpler models like Lasso (l_1-regularized) or RF regression, which utilize principal component features derived from linguistic features and clinical embeddings, outperform more complex, fine-tuned NLP models. In the winner announcement, the RF model secured the 4th position out of 43 submissions for difficulty, while the Lasso model attained the 2nd position out of 34 for response time. Post-hoc analyses revealed that if we had submitted the predictions of the Lasso model of difficulty to the leaderboard, we would have surpassed the second position in predicting difficulty as well.

Moreover, predicting the response time for medical multiple-choice questions has proven to be a more straightforward task compared to predicting the difficulty of such questions. Response time primarily hinges on item length (i.e., word count and number of answers), item type, exam step, and the level of analytical thinking required for the answers, as illustrated by permutation feature importance analyses. Conversely, predicting item difficulty poses greater challenges, with all models approaching an average baseline performance. Nevertheless, post-hoc analyses suggest that more extensive experimentation with fine-tuned language models could potentially aid in discerning the difficulty of multiple-choice questions. While response time can be more accurately predicted from linguistic features like word count, predicting difficulty may require more intricate modeling of deep clinical text representations.

6 Limitations

In the future study, we could deepen our understanding of our findings, potentially shedding light on the circumstances in which simpler models might be advantageous.

One initial limitation we would have liked to tackle is the utilization of student responses instead of percentage- and mean-aggregated targets. This limitation stems from the fact that we only received aggregated or summarized data for difficulty and response time per item, rather than the individual-level data. Access to the individual-level data would have allowed us to explore more advanced psychometric models that consider interactions between items and students.

Another limitation we aim to address is conducting a more comprehensive study on fine-tuning language models. Specifically, we plan to delve into a more exhaustive grid search, which could potentially illuminate the most optimal model initialization and hyperparameters.

Finally, another constraint of our study is the possibility of overlooked features in the data. This limitation arises from our focus on a predetermined set of features, including LIWC, TAALES, and the BERT clinical model, for feature selection. In future research, additional methods for feature extraction could be explored.

References

aaditya. 2022. Bio_clinicalbert_emrqa. https://huggingface.co/aaditya/Bio_ClinicalBERT_emrqa. Accessed: 03/14/2024.

H. Akaike. 1974. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723.

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly Available Clinical BERT Embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Ryan L. Boyd, Ashwini Ashokkumar, Shiva Seraj, and James W. Pennebaker. 2022. The development and psychometric properties of liwc-22.

exafluence. 2021. Bert-clinicalqa. https://huggingface.co/exafluence/BERT-ClinicalQA. Accessed: 03/14/2024.

Le An Ha, Victoria Yaneva, Peter Baldwin, and Janet Mee. 2019a. Predicting the Difficulty of Multiple Choice Questions in a High-stakes Medical Exam. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 11–20, Florence, Italy. Association for Computational Linguistics.

Le An Ha, Victoria Yaneva, Peter Baldwin, and Janet Mee. 2019b. Predicting the difficulty of multiple choice questions in a high-stakes medical exam. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 11–20, Florence, Italy. Association for Computational Linguistics.

Kristopher Kyle and Scott A. Crossley. 2015. Automatically assessing lexical sophistication: Indices, tools, findings, and application. *TESOL Quarterly*, 49(4):757–786.

Yikuan Li, Ramsey M Wehbe, Faraz S Ahmad, Hanyin Wang, and Yuan Luo. 2022. Clinical-longformer and clinical-bigbird: Transformers for long clinical sequences. *arXiv preprint arXiv:2201.11838*.

Yikuan Li, Ramsey M Wehbe, Faraz S Ahmad, Hanyin Wang, and Yuan Luo. 2023. A comparative study of pretrained language models for long clinical text. *Journal of the American Medical Informatics Association*, 30(2):340–347.

Andrey Lovakov and Elena R. Agadullina. 2021. Empirically derived guidelines for effect size interpretation in social psychology. *European Journal of Social Psychology*, 51:485–504.

Ronald Ortner and Hannes Leitgeb. 2011. Mechanizing induction. In Dov M. Gabbay, Stephan Hartmann, and John Woods, editors, *Inductive Logic*, volume 10 of *Handbook of the History of Logic*, pages 719–772. North-Holland.

James W. Pennebaker, Ryan L. Boyd, Roger J. Booth, Ashwini Ashokkumar, and Martha E. Francis. 2022. Linguistic inquiry and word count: Liwc-22. *Pennebaker Conglomerates*.

James W. Pennebaker, Cindy K. Chung, Joey Frazee, Gary M. Lavergne, and David I. Beaver. 2014. When small words foretell academic success: The case of college admissions essays. *PLoS ONE*, 9(12):e115844.

Gideon Schwarz. 1978. Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464.

W. N. Venables and B. D. Ripley. 2002. *Modern Applied Statistics with S*. Springer New York.

Guangyu Wang, Xiaohong Liu, Zhen Ying, Guoxing Yang, Zhiwei Chen, Zhiwen Liu, Min Zhang, Hongmei Yan, Yuxing Lu, Yuanxu Gao, et al. 2023. Optimized glycemic control of type 2 diabetes with reinforcement learning: a proof-of-concept trial. *Nature Medicine*, 29(10):2633–2642.

David H Wolpert and William G Macready. 1997. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82.

Kang Xue, Victoria Yaneva, Christopher Runyon, and Peter Baldwin. 2020. Predicting the Difficulty and Response Time of Multiple Choice Questions Using Transfer Learning. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 193–197, Seattle, WA, USA → Online. Association for Computational Linguistics.

Victoria Yaneva, Le An Ha, Peter Baldwin, and Janet Mee. 2020. Predicting Item Survival for Multiple Choice Questions in a High-Stakes Medical Exam. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6812–6818, Marseille, France. European Language Resources Association.

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.

Victoria Yaneva, Kai North, Peter Baldwin, An Ha Le, Saed Rezayi, Yiyun Zhou, Sagnik Ray Choudhury, Polina Harik, and Brian Clauser. 2024. Findings from the First Shared Task on Automated Prediction of Difficulty and Response Time for Multiple Choice Questions. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Item Difficulty and Response Time Prediction with Large Language Models: An Empirical Analysis of USMLE Items

Okan Bulut, Guher Gorgun, Bin Tan
Measurement, Evaluation, and Data Science
Faculty of Education, University of Alberta, Canada
{bulut, gorgun, btan4}@ualberta.ca

Abstract

This paper summarizes our methodology and results for the BEA 2024 Shared Task. This competition focused on predicting item difficulty and response time for retired multiple-choice items from the United States Medical Licensing Examination® (USMLE®). We extracted linguistic features from the item stem and response options using multiple methods, including the BiomedBERT model, FastText embeddings, and Coh-Metrix. The extracted features were combined with additional features available in item metadata (e.g., item type) to predict item difficulty and average response time. The results showed that the BiomedBERT model was the most effective in predicting item difficulty, while the fine-tuned model based on FastText word embeddings was the best model for predicting response time.

1 Introduction

In standardized exams, the examination of item characteristics is highly crucial for ensuring the fairness and validity of test results. For example, the difficulty of items pertains to the likelihood of an examinee answering the items correctly. Incorporating a broad range of item difficulty levels in a standardized exam can help reduce measurement error and thereby improve the accuracy of the measurement process (Kubiszyn and Borich, 2024). In addition, while response time is often linked to item difficulty (i.e., more difficult items require more time to answer) (Yang et al., 2002), this variable itself can also offer new insights into examinees' test completion processes, such as their testing engagement and cognitive processes, thereby supporting the validity of test results. Furthermore, understanding item characteristics can also be advantageous for modern test administration methods, including applications in automated item assembly, computerized adaptive testing, and personalized assessments (Baylari and Montazer, 2009; Wauters et al., 2012).

The difficulty of items and the average response time required to answer them are typically estimated based on empirical data collected during test pretesting. However, pretesting and obtaining robust results often require a large sample of examinees, which can incur substantial test administration costs. As a result, researchers have explored various methods to predict item characteristics without an actual test administration. For instance, researchers have sought estimates of item difficulty from domain experts and test development professionals. However, this approach has not consistently produced satisfactory or reliable estimations (Bejar, 1983; Attali et al., 2014; Wauters et al., 2012; Impara and Plake, 1998). Another line of research seeks to predict item characteristics based on only item texts, such as the passages in source-based items, item stem, and response options (Yaneva et al., 2019; Hsu et al., 2018). This approach employs text-mining techniques to extract surface features (e.g., the number of words in the texts) and complex features (e.g., semantic similarities of sentences) from item texts, to make predictions using advanced statistical models.

Building on the second line of research in predicting item characteristics based on item texts, the National Board of Medical Examiners (NBME) initiated the BEA 2024 Shared Task (`https://sig-edu.org/sharedtask/2024`) for automated prediction of item difficulty and item response time. The released dataset contained 667 previously used and now retired items from the United States Medical Licensing Examination® (USMLE®). The USMLE is a series of high-stakes examinations (also known as Steps; `https://www.usmle.org/step-exams`) to support medical licensure decisions in the United States. The items from USMLE Steps 1, 2 Clinical Knowledge (CK), and 3 focus on a wide range of topics relevant to the practice of medicine.

In the BEA 2024 Shared Task, research teams

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 522–527
June 20, 2024 ©2024 Association for Computational Linguistics

were invited to utilize natural language processing (NLP) methods for extracting linguistic features of the items and using them to predict the difficulty and response time of the items. Our team employed state-of-the-art large language models (LLMs) to extract the features and build predictive models for item difficulty and response time. This paper documents the methods and results of our best-performing models for predicting item difficulty and response time separately.

2 Related work

The interest and effort in predicting item difficulty based on item texts dates back decades in the measurement literature. Early work in item difficulty prediction primarily focused on identifying how item difficulty is influenced by a set of readily available, easily extracted, or manually coded item-level features. For example, Drum et al. (1981) predicted the difficulty of 210 reading comprehension items using various surface structure variables and word frequency measures for the text, such as the number of words, content words, or content-function words. Freedle and Kostin (1993) predicted the difficulty of 213 reading comprehension items using 12 categories of sentential and discourse variables, such as vocabulary, length of texts, and syntactic structures (e.g., the number of negations). Perkins et al. (1995) employed artificial neural networks to predict the item difficulty of 29 items in a reading comprehension test. They coded the items to extract three types of features: text structure (e.g., the number of words, lines, paragraphs, sentences, and content words), propositional analysis of passages and stems (e.g., the number of arguments, modifiers, and predicates), and cognitive process (e.g., identify, recognize, verify, infer, generalize, or problem-solving).

Research focused on the prediction of item characteristics such as difficulty and response time has been significantly influenced by the availability and application of emerging techniques in NLP and machine learning AlKhuzaey et al. (2023). For example, Yaneva et al. (2019) employed NLP methods to extract syntactic features to predict item difficulty, which were identified as crucial predictors. Another application of NLP methods involves assessing the linguistic complexity or readability of item texts to predict item difficulty. Benedetto et al. (2020a), for instance, calculated readability indices for item texts and combined them with other features to predict item difficulty. However, readability indices did not perform well as predictors of item difficulty–a finding consistent with Susanti et al. (2017) who noted that readability indices were among the least important predictors of item difficulty.

NLP methods can also be used to extract Term Frequency-Inverse Document Frequency (TF-IDF) features. TF-IDF measures the frequency of words or word sequences in a document and adjusts this count based on their frequency across a collection of documents. This approach emphasizes the importance of specific words to a particular document, with higher values indicating greater potential importance (Salton, 1983). In a relatively recent study predicting item difficulty for newly generated multiple-choice questions, Benedetto et al. (2020b) extracted TF-IDF features and achieved a root mean square error of 0.753.

An important application of NLP techniques is the extraction of semantic features from item texts. Word embedding is a technique that converts texts into numerical values in vector space, capturing the meanings of words across different dimensions (Mikolov et al., 2013). Pre-trained NLP models such as Word2Vec and GLoVe allow researchers to extract word embedding features from item texts (e.g., Firoozi et al., 2022). For example, Hsu et al. (2018) transformed item texts into semantic vectors and then used cosine similarity to measure the semantic similarity between different pairs of items. Additionally, (Yaneva et al., 2019) extracted word embedding features from multiple-choice items in high-stakes medical exams. Along with other linguistic and psycholinguistic features in predicting item difficulty, they found that word embedding features contributed most to the predictive power.

More recently, a significant breakthrough in the NLP field has been the development of LLMs such as BERT (Devlin et al., 2018) and its variants, which were trained using different mechanisms or training datasets. For example, Zhou and Tao (2020) utilized a BERT-variant model to predict the difficulty of programming problems. Their results showed that compared with BERT, DistilBERT, a small version of the BERT base model, was the best-performing model when the only available data for fine-turning was the text of the items. Benedetto et al. (2021) also compared the performance of BERT and DistilBERT in predicting the difficulty of multiple-choice questions and found that the BERT-based models significantly outperformed the two baseline models.

Unlike the prediction of item difficulty, the prediction of response time has not been widely investigated in the literature. This is mainly due to the limited availability of response time data. However, with the increasing use of digital assessments, such as computer-based and computerized-adaptive tests, in operational testing, the collection of response data has become easier, which motivated researchers to employ predictive models to predict the average response time required to solve the items (e.g., Baldwin et al., 2021; Hecht et al., 2017; Yaneva et al., 2019).

3 Methodology

3.1 Dataset

As mentioned earlier, this study utilized an empirical dataset released by NBME, which included 667 multiple-choice items previously administered in the USMLE series. Due to the requirements of the BEA 2024 Shared Task, the data was released in two stages. Initially, 446 multiple-choice items were provided for extracting linguistic features from the items and building predictive models based on the extracted features. For each item, the dataset encompasses the source texts (typically a clinical case followed by a question) and the texts for each response option. The response options for the questions vary and can include up to 10 options, each represented in a separate column. When the number of response options was less than 10, the remaining columns were left empty.

Additionally, the dataset contained metadata with four additional variables: Item type (text-only items versus items containing pictures), exam steps (Steps 1, 2, or 3 in the USMLE series), item difficulty, and average response time. Subsequently, the predictive models trained in the first stage were applied to make predictions for the remaining 201 items in the second stage, serving as the testing set for evaluating the performance of the predictive models for item difficulty and response time. The structure of the second dataset mirrors that of the first, with the exception that the item difficulty and response time variables were not immediately available. These variables were released after the submission deadline for the BEA 2024 Shared Task, allowing for the identification of the best-performing trials among the participating teams.

3.2 Our Best Model for Difficulty Prediction

Here, we describe the details of our best-performing model for predicting item difficulty ($RMSE = .318$), which performed slightly worse than a baseline dummy regressor ($RMSE = .311$) and ranked at the 20th place out of 43 submissions in the difficulty prediction leaderboard.

3.2.1 Feature Extraction

We extracted linguistic features from item stems and response alternatives (i.e., the answer key and the incorrect response options) by leveraging both pre-trained large-language models and more interpretable text representations such as connectivity, cohesion, and text length. We started the feature extraction process by concatenating the stem, key, and alternatives of each item in a single data frame column and separated each item into individual data files to extract Coh-Metrix features (McNamara et al., 2014; Graesser et al., 2011). Concatenating item stems and alternatives served two purposes: (1) Adequately represent item length in terms of stem and alternatives and (2) control for the differential number of alternatives that each item includes. Coh-Metrix includes 108 features and analyzes a text on multiple measures of language and discourse (Graesser et al., 2011).

Coh-Metrix focuses on six theoretical levels of text representation: words, syntax, the explicit textbase, the referential situation model, the discourse genre and rhetorical structure, and the pragmatic communication level (Graesser et al., 2014). It generates indices of text, including paragraph count, sentence count, word count, narrativity, syntactic simplicity, referential cohesion, deep cohesion, noun overlap, stem overlap, latent semantic analysis, lexical diversity, syntactic complexity, syntactic pattern density, and readability. We removed four features from Coh-Metrix indices due to no variability, including paragraph count (i.e., the number of paragraphs), the standard deviation of paragraph length, the mean Latent Semantic Analysis (LSA) overlap in adjacent paragraphs, and the standard deviation of LSA overlap in adjacent paragraphs.

In the next step, we utilized the BiomedBERT model (Gu et al., 2020) to extract new features. This model, which was previously named PubMedBERT, is a pretrained LLM based on abstracts from PubMed and full-text articles from PubMedCentral. We chose this particular model because it is known to achieve state-of-the-art performance on many biomedical NLP tasks. By using BiomedBERT, we obtained sentence embeddings for the item stems and alternatives and then computed the cosine similarity between item sentence embeddings and alternative stem embeddings. Cosine similarity, which is commonly used to quantify the degree of similarity between two sets of information, was computed as the cosine angle between the embedding vectors of item stem and alternatives. As cosine similarity, ranging between 0 and 1, gets closer to 1, it indicates more resemblance between the embedding vectors obtained using the item stem and alternatives.

In the final step, we also extracted word embeddings for the concatenated text using stems and alternatives by tokenizing the text using the BiomedBERT model (Gu et al., 2020). BiomedBERT has 768 dimensions with a maximum length of 512 words. We extracted the last hidden layer of embeddings. We created a new data frame composed of three sets of features extracted (i.e., Coh-Metrix features using the stem, key, and alternatives of each item, the cosine similarity between the stem and alternatives, and word embeddings using the stem, key, and alternatives) and the ground truth of item difficulty. The final data frame is composed of 882 features and the target variable of item difficulty.

3.2.2 Model Training

To identify the best model with the lowest RMSE value, we used 85% of the data as our training set and 15% as our holdout test set. Because the sample size was too small ($N = 466$ of items shared in total) and we had a very large set of features (*N = 882*), we first applied a dimension reduction technique, *Principal Component Analysis* (PCA) (Wold et al., 1987). A PCA model with 30 components explained 99% of variability in the dataset, and thus, the final feature set included 30 components extracted through the PCA analysis. We used lasso regression (Tibshirani, 1996) with repeated 5-fold cross-validation to select the best hyperparameter (i.e., *alpha*). *Alpha* in lasso regression is the model penalty that determines the amount of shrinkage in the model. An advantage of lasso regression is the application of a regularization algorithm that controls for the irrelevant features in the model by shrinking the contribution of irrelevant features to zero. An alpha value of .01 yielded the best model during the cross-validation stage.

3.2.3 Results

With our pseudo-test set held out from the shared training set, we obtained a Mean Squared Error (MSE) value of .064, a Root Mean Squared Error (RMSE) value of .253, and a Mean Absolute Error (MAE) value of .190, and a Pearson's correlation coefficient of .555.

3.3 Our Best Model for Response Time Prediction

Our solution that achieved the best performance in predicting response time differed from the one that was best at predicting item difficulty. This solution is briefly documented below.

3.3.1 Feature Extraction

First, FastText word embeddings were generated for each item stem and response option. We employed the pre-trained FastText embeddings (wiki-news-300d-1m.vec.zip; obtained from https://fasttext.cc/docs/en/english-vectors.html) to map each word in the text to its corresponding 300-dimensional vector representation. FastText is a modified version of word2vec; the difference is that it treats each word as composed of n-grams rather than the original word in Word2Vec (Mikolov et al., 2017). For each text option, the embeddings of the first 60 words were concatenated to form a feature vector, resulting in a dimension of 18,000 (60 words × 300 dimensions) for each option. If the text had fewer than 60 words, the corresponding vector was padded with zeros.

Similar to the approach taken for item difficulty predictions, cosine similarity scores between each pair of alternatives (i.e., response options) were calculated using the embeddings from the BiomedBERT model. For each pair, the cosine similarity between their embeddings was computed to capture the semantic differences between different response options. The extracted features were then combined with the dummy-coded item development information (e.g., text-based items only vs. items including pictures; administration step in the USMLE series) to form the final feature set. Unlike in the item difficulty prediction, we did not extract any other linguistic features in response time prediction.

3.3.2 Model Training

Considering the extremely high dimensionality of the features, we performed feature selection and dimension reduction techniques. First, using the available information on response time in the training set ($N = 466$), we eliminated the feature columns that had an absolute correlation coefficient smaller than 0.1. Then, we performed PCA to extract principal components until they could capture 95% of the information presented in the original feature set. To this end, we obtained a final feature set with 339 features to train an algorithm.

As before, the model training involved the use of lasso regression due to its ability to perform feature selection and handle multicollinearity in high-dimensional data. The training process was performed using 10-fold cross-validation to optimize the hyperparameter (i.e., alpha) and evaluate the model's performance. In terms of the hyperparameter search space, the regularization strength (*alpha*) was tuned using a randomized search over a logarithmic scale from 1e-4 to 1e-0.05, with 1000 candidate values. An alpha value of .44 yielded the best model during the cross-validation stage. Additionally, the *fit intercept* parameter was tested with both True and False values, while the *selection parameter* was tested with 'cyclic' and 'random' options[1].

3.3.3 Results

Upon comparing our predicted response time and the released response time from the BEA 2024 Shared Task, we found this solution ($RMSE = 31.48$; $MSE = 990.98$; $MAE = 23.54$, $r = 0.209$) was slightly better than the baseline dummy regressor ($RMSE = 31.68$), which ranked 24th among the 34 submissions.

4 Discussion and Conclusion

The competition results for the BEA 2024 Shared Task indicated that it is difficult to predict item characteristics such as difficulty using linguistic features (Yaneva et al., 2024). Only 15 teams out of 43 managed to perform better than a baseline dummy regressor when it comes to predicting item difficulty using textual features extracted from the items. These results suggest that linguistic features may not be sufficient to capture the complex interplay between item features and item difficulty.

Unlike predicting item difficulty, predicting the average response time using linguistic features appears to be a more promising task. Out of 34 submissions, 24 teams performed better than a baseline dummy regressor in predicting the average response time. This finding is not necessarily surprising because the average reading time required for the items is likely to be correlated with the linguistic features extracted from the items.

Overall, the results for the BEA 2024 Shared Task indicate that predicting item characteristics such as difficulty remains challenging and requires factors beyond linguistic or textual features.

[1] Our codes for predicting item difficulty and response time are available at `https://osf.io/dwe4n/`.

References

Samah AlKhuzaey, Floriana Grasso, Terry R Payne, and Valentina Tamma. 2023. Text-based question difficulty prediction: A systematic review of automatic approaches. *International Journal of Artificial Intelligence in Education*, pages 1–53.

Yigal Attali, Luis Saldivia, Carol Jackson, Fred Schuppan, and Wilbur Wanamaker. 2014. Estimating item difficulty with comparative judgments. *ETS Research Report Series*, 2014(2):1–8.

Peter Baldwin, Victoria Yaneva, Janet Mee, Brian E Clauser, and Le An Ha. 2021. Using natural language processing to predict item response times and improve test construction. *Journal of Educational Measurement*, 58(1):4–30.

Ahmad Baylari and Gh A Montazer. 2009. Design a personalized e-learning system based on item response theory and artificial neural network approach. *Expert Systems with Applications*, 36(4):8013–8021.

Isaac I Bejar. 1983. Subject matter experts' assessment of item statistics. *Applied Psychological Measurement*, 7(3):303–310.

Luca Benedetto, Giovanni Aradelli, Paolo Cremonesi, Andrea Cappelli, Andrea Giussani, and Roberto Turrin. 2021. On the application of transformers for estimating the difficulty of multiple-choice questions from text. In *Proceedings of the 16th workshop on innovative use of NLP for building educational applications*, pages 147–157.

Luca Benedetto, Andrea Cappelli, Roberto Turrin, and Paolo Cremonesi. 2020a. Introducing a framework to assess newly created questions with natural language processing. In *International Conference on Artificial Intelligence in Education*, pages 43–54. Springer.

Luca Benedetto, Andrea Cappelli, Roberto Turrin, and Paolo Cremonesi. 2020b. R2de: a nlp approach to estimating irt parameters of newly generated questions. In *Proceedings of the tenth international conference on learning analytics & knowledge*, pages 412–421.

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*.

Priscilla A Drum, Robert C Calfee, and Linda K Cook. 1981. The effects of surface structure variables on performance in reading comprehension tests. *Reading Research Quarterly*, pages 486–514.

T Firoozi, O Bulut, C Demmans Epp, A N Abadi, and D Barbosa. 2022. The effect of fine-tuned word embedding techniques on the accuracy of automated essay scoring systems using neural networks. *Journal of Applied Testing Technology*, 23:21–29.

Roy Freedle and Irene Kostin. 1993. The prediction of toefl reading item difficulty: Implications for construct validity. *Language Testing*, 10(2):133–170.

Arthur C Graesser, Danielle S McNamara, Zhiqang Cai, Mark Conley, Haiying Li, and James Pennebaker. 2014. Coh-metrix measures text characteristics at multiple levels of language and discourse. *The Elementary School Journal*, 115(2):210–229.

Arthur C Graesser, Danielle S McNamara, and Jonna M Kulikowich. 2011. Coh-metrix: Providing multilevel analyses of text characteristics. *Educational Researcher*, 40(5):223–234.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2020. Domain-specific language model pretraining for biomedical natural language processing.

Martin Hecht, Thilo Siegle, and Sebastian Weirich. 2017. A model for the estimation of testlet response time to optimize test assembly in paper-and-pencil large-scale assessments. *Journal for educational research online*, 9(1):32–51.

Fu-Yuan Hsu, Hahn-Ming Lee, Tao-Hsing Chang, and Yao-Ting Sung. 2018. Automated estimation of item difficulty for multiple-choice tests: An application of word embedding techniques. *Information Processing & Management*, 54(6):969–984.

James C Impara and Barbara S Plake. 1998. Teachers' ability to estimate item difficulty: A test of the assumptions in the angoff standard setting method. *Journal of Educational Measurement*, 35(1):69–81.

Tom Kubiszyn and Gary D Borich. 2024. *Educational testing and measurement*. John Wiley & Sons.

Danielle S McNamara, Arthur C Graesser, Philip M McCarthy, and Zhiqiang Cai. 2014. *Automated evaluation of text and discourse with Coh-Metrix*. Cambridge University Press.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2017. Advances in pre-training distributed word representations. *arXiv preprint arXiv:1712.09405*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Kyle Perkins, Lalit Gupta, and Ravi Tammana. 1995. Predicting item difficulty in a reading comprehension test with an artificial neural network. *Language testing*, 12(1):34–53.

Gerard Salton. 1983. Introduction to modern information retrieval. *McGraw-Hill*.

Yuni Susanti, Takenobu Tokunaga, Hitoshi Nishikawa, and Hiroyuki Obari. 2017. Controlling item difficulty for automatic vocabulary question generation. *Research and practice in technology enhanced learning*, 12:1–16.

Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288.

Kelly Wauters, Piet Desmet, and Wim Van Den Noortgate. 2012. Item difficulty estimation: An auspicious collaboration between data and judgment. *Computers & Education*, 58(4):1183–1193.

Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52.

Victoria Yaneva, Peter Baldwin, Janet Mee, et al. 2019. Predicting the difficulty of multiple choice questions in a high-stakes medical exam. In *Proceedings of the fourteenth workshop on innovative use of NLP for building educational applications*, pages 11–20.

Victoria Yaneva, Kai North, Peter Baldwin, An Ha Le, Saed Rezayi, Yiyun Zhou, Sagnik Ray Choudhury, Polina Harik, and Brian Clauser. 2024. Findings from the First Shared Task on Automated Prediction of Difficulty and Response Time for Multiple Choice Questions. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Chien-Lin Yang, Thomas R O Neill, and Gene A Kramer. 2002. Examining item difficulty and response time on perceptual ability test items. *Journal of Applied Measurement*, 3(3):282–299.

Ya Zhou and Can Tao. 2020. Multi-task bert for problem difficulty prediction. In *2020 international conference on communications, information system and computer engineering (cisce)*, pages 213–216. IEEE.

Utilizing Machine Learning to Predict Question Difficulty and Response Time for Enhanced Test Construction

Rishikesh Fulari
Purdue University, Fort Wayne
fularp01@pfw.edu

Jonathan Rusert
Purdue University, Fort Wayne
jrusert@pfw.edu

Abstract

In this paper, we present the details of our contribution to the BEA Shared Task on Automated Prediction of Item Difficulty and Response Time. Participants in this collaborative effort are tasked with developing models to predict the difficulty and response time of multiple-choice items within the medical domain. These items are sourced from the United States Medical Licensing Examination® (USMLE®), a significant medical assessment. In order to achieve this, we experimented with two featurization techniques, one using lingusitic features and the other using embeddings generated by BERT fine-tuned over MS-MARCO dataset. Further, we tried several different machine learning models such as Linear Regression, Decision Trees, KNN and Boosting models such as XGBoost and GBDT. We found that out of all the models we experimented with Random Forest Regressor trained on Linguistic features gave the least root mean squared error, securing fourteenth rank out of 43 for Item Difficulty Prediction and ninth rank out of 34 for Response Time Prediction. We made our code publicly available on GitHub.[1].

1 Introduction

To conduct fair standardized tests for evaluating the learning outcomes of students, it is necessary to design tests that cover variety of questions of all difficulty levels such as 'easy', 'moderate' and 'difficult' ones. Allowed exam time is another component that impacts the difficulty of exam. Allowing ample amount of time to solve the questions can considerably reduce the difficulty whereas providing very little time to solve the exam questions can on other hand, make the exam unreasonably difficult. Thus, the difficulty level of questions and the time taken to solve the questions(response time) are two critical factors to determine the overall difficulty of exam.

Determining the difficulty of items as well as the response time for this task, is a challenge in itself. Conventionally, item difficulty and the response time are gathered through pretesting, where new items are incorporated into live exams alongside scored items. However, this process is labor-intensive and costly, often limiting the number of items that can be created. Furthermore, the reliance on pretesting poses security risks, as items may be copied or leaked due to their repeated usage.

To tackle these challenges, there's a growing interest in predicting item characteristics such as difficulty and response time directly from the item text. This approach, known as the "cold-start parameter estimation problem" (McCarthy et al., 2021) aims to streamline the process and enhance fairness by reducing the reliance on pretesting. By utilizing predictive models, estimates of item difficulty and response time can be generated, enabling a more efficient parameter estimation process with a smaller sample of test-takers.

In this paper, we examine several approaches which build on predictive machine learning models (for example, linear regression, decision trees) and deep learning models(such as BERT). Our best model for the task of item difficulty achieved RMSE of 0.31 and the best model for the task of predicting response time achieved RMSE of 31.68. We hope that the exploration of models in this paper is able to help future researchers in the evaluation of exams.

2 Related Work

One of the earliest applications of predicting item difficulty emerged in the realm of language testing. Here, a framework was introduced to assess learners' language proficiency in English, German, or French (Beinborn et al., 2015). Controlling the

[1] https://github.com/rishikeshF/sig-edu-bea-2024-predicting-response-time-and-question-difficulty

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 528–533
June 20, 2024 ©2024 Association for Computational Linguistics

difficulty of tests has also been important for automated generation of MCQ format tests(Alsubait et al., 2013). Another application can be found in context of automated grading where question difficulty estimates guide test creation(Padó, 2017). Thus, predicting item difficulty has been a subject of growing research and with passage of time has extended to high-stakes applications such as medical or clinical exam(Yaneva et al., 2020).

In order to automate difficulty prediction, machine learning and NLP based approaches using word lengths, sentence lengths and tf-idf featurization were proposed (Settles et al., 2020). A further improvement to it can be seen in the form of introduction of linguistic features (Yaneva et al., 2021) which drastically improved the performance of approaches based on using machine learning models.

On similar lines, machine learning and deep learning approaches have been researched upon for predicting the response time (Baldwin et al., 2021). Other techniques employed in this regard include using transfer learning (Xue et al., 2020) and language models such as BERT (Devlin et al., 2019).

3 Experiments

For assessing performance, Root Mean Squared Error (RMSE) serves as the metric for predicting response times and item difficulty in regression tasks. Its suitability stems from the intuitive nature of RMSE, making it well-aligned with the nature of these regression tasks.

3.1 Dataset

The data for both tasks, response time prediction and difficulty prediction, consists of 667 previously used and now retired Multiple Choice Questions (MCQs) from USMLE Steps 1, 2 CK, and 3. The USMLE is a series of examinations (called Steps) to support medical licensure decisions in the United States that is developed by the National Board of Medical Examiners (NBME) (Yaneva et al., 2024) and Federation of State Medical Boards (FSMB). Here is a sample question from USMLE Step 1.

> Q. A 65-year-old woman comes to the physician for a follow-up examination after blood pressure measurements were 175/105 mm Hg and 185/110 mm Hg 1 and 3 weeks ago, respectively. She has well-controlled type 2 diabetes mellitus. Her blood pressure now is 175/110 mm Hg. Physical examination shows no other abnormalities. Antihypertensive therapy is started, but her blood pressure remains elevated at her next visit 3 weeks later. Laboratory studies show increased plasma renin activity; the erythrocyte sedimentation rate and serum electrolytes are within the reference ranges. Angiography shows a high-grade stenosis of the proximal right renal artery; the left renal artery appears normal. Which of the following is the most likely diagnosis?
> (A) Atherosclerosis
> (B) Congenital renal artery hypoplasia
> (C) Fibromuscular dysplasia
> (D) Takayasu arteritis
> (E) Temporal arteritis

The part describing the case is referred to as stem, the correct answer is referred to as key, and the incorrect answer options are known as distractors. All items are MCQs that test medical knowledge and were written by experienced subject matter experts following a set of guidelines, stipulating adherence to a standard structure. These guidelines require avoidance of “window dressing” (extraneous material not needed to answer the item), “red herrings” (information designed to mislead the test-taker), and grammatical cues (e.g., correct answers that are longer or more specific than the other options). The goal of standardizing items in this manner is to produce items that vary in their difficulty and discriminating power due only to differences in the medical content they assess. The items were administered within a standard nine-hour exam. For this shared task, the item characteristic data was derived from first-time examinees from accredited US and Canadian medical schools.

Each item is tagged with the following item characteristics:

- **Item difficulty** A measure of item difficulty where higher values indicate more difficult items.

- **Time intensity** Arithmetic mean response time, measured in seconds, across all examinees who attempted a given item in a live exam. This includes all time spent on the item from the moment it is presented on the screen until the examinee moves to the next item, as well as any revisits.3. Feature engineering

- **ItemNum** denotes the consecutive number of the item in the dataset (e.g., 1,2,3,4,5, etc).
- **ItemStem_Text** contains the text data for the item stem (the part of the item describing the clinical case).
- **Answer_A** contains the text for response option A
- **Answer_B** contains the text for response option B
- **Answer_C** contains the text for response option C.

  (...)
- **Answer_J** contains the text for response option J. For items that have fewer than J response options, the remaining columns are left blank. For example, if an item contains response options A to E, the fields for columns F to J are left blank for that item.
- **Answer_Key** contains the letter of the correct answer for that item.
- **Answer_Text** contains the text of the correct response for the item.
- **ItemType** denotes whether the item contained an image (e.g., an x-ray image, picture of a skin lesion, etc.) or not. The value "Text" denotes text-only items that do not contain images and the value "PIX" denotes items that contain an image. Note that the images are not part of the dataset.
- **EXAM** denotes the Step of the USMLE exam the item belongs to (Step 1, Step 2, or Step 3). For more information on the Steps of the USMLE see https://www.usmle.org/step-exams.
- **Difficulty** contains the item difficulty measure. Higher values indicate more difficult items.
- **Response_Time** contains the mean response time for the item measured in seconds.

The training set comprised of 466 examples and the test set contained 201 items. The combined length of question, multiple choices and the answer was mostly less than 200 words and maxed out at 379. Figure 1 shows a distribution on the number of words in the examples.

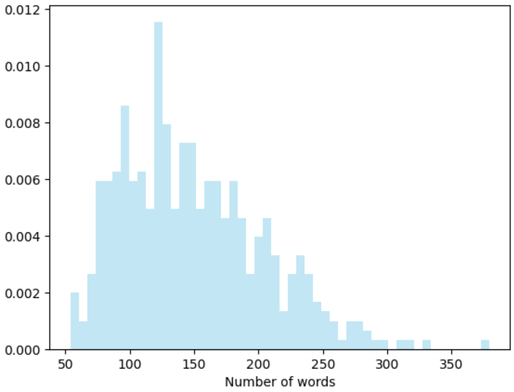


Figure 1: Number of words histogram

4 Methods

4.1 Feature Engineering

For both the tasks, two featuring engineering approaches were tried. First, using embeddings the entire text (comprised of question, multiple choices and corresponding answer) was converted into a 768 dimensional vector. The second approach used linguistic features. The details of both the approaches are as follows:

4.1.1 Embeddings

In order to represent the textual features, sentence Transformer based embeddings were used. The sentence transformer model used is pritamdeka/S-PubMedBert-MS-MARCO (Deka et al., 2022) (from HuggingFace). It maps sentences and paragraphs to a 768 dimensional dense vector space and can be used for tasks like clustering or semantic search. This sentence transformer model has been developed by fine-tuning microsoft/BiomedNLP-PubMedBERT-base-uncased-abstract-fulltext (Gu et al., 2020) model on MS-MARCO (Nguyen et al., 2016) dataset. It can be used for the information retrieval task in the medical or health domain.

4.1.2 Text based/Linguistic Features

Another method explored for text representation involved leveraging specific linguistic features: word count, number of unique words, number of additives, number of unique additives, number of normalized additives, as well as counts of numbers and letters. Additionally, two additional features, namely 'ItemType' (indicating the presence of a picture) and 'EXAM' (exam level), were incorporated. Each question and word were consequently encoded into a nine-dimensional vector, encompassing these precise linguistic characteristics for subsequent analysis.

4.2 Machine Learning Models

Various machine learning models were explored, encompassing classical approaches like linear regression, Decision Tree Regressor, and K-Nearest Neighbours regressor, as well as advanced techniques such as fine-tuned language models like BERT, simple one-neuron networks, and ensemble methods including random forest regressor. Additionally, boosted models like gradient boosted decision trees regressor and XGBoost regressor were investigated.

4.2.1 Hyperparameters for difficulty prediction

- **Decision Trees** Max-depth: 3
- **KNN** Number of neighbors: 7
- **XGBRegressor** Max-depth: 5

  Number of estimators: 700
- **GBDT**: Max-depth: 3

  Number of estimators: 600
- **Random Forest Regressor**: Max-depth: 3

  Number of estimators: 700

4.2.2 Hyperparameters for response time prediction

- **Decision Trees** Max-depth: 3
- **KNN** Number of neighbors: 7
- **XGBRegressor** Max-depth: 5

  Number of estimators: 600
- **GBDT**: Max-depth: 3

  Number of estimators: 600
- **Random Forest Regressor** Max-depth: 6

  Number of estimators: 800

5 Observations and Results

Table 1 depicts the Root Mean Squared Error obtained for different machine learning models and neural networks along with the corresponding featurization method used.

In our investigation employing various machine learning models and featurization techniques, we found Fine-Tuned BERT to yield consistently stable results, with the lowest RMSE of 0.31 in the task of difficulty prediction. Conversely, our analysis revealed that the Random Forest Regressor, particularly when paired with Linguistic Features, exhibited superior performance in predicting response time with RMSE of 31.68. These results were based on the models trained on a subset of training dataset instead of entire training set, as a smaller subset was used for validation and testing prior to the release of test set.

After training the models on entire training data, the results obtained differed from the previous results. This time, Linguistic features used with Linear Regression gave the lowest RMSE of 0.302 on Difficulty prediction and RMSE of 26.181 on Response Time prediction. These were closely matched by Random Forest Regressor with scores of 0.303 and 26.234 for Difficulty prediction and Response Time prediction respectively. Table 1 displays the RMSE obtained for different models.

A noteworthy observation here is that Linear Regression performed the worst(RMSE of 0.614) with embeddings as features for Difficulty prediction task but performed the best(RMSE of 0.302) when used with Linguistic features, surpassing all other models. This substantial improvement in the performance can be attributed to the fact that total number of input vector size was reduced from 768 dimensions(when used with embeddings) to 9 dimensions(when used with linguistic features), thus eliminating the 'curse of dimensionality'.

Notably, linguistic features, encompassing syntactic aspects such as word count and the presence of additives, emerged as pivotal predictors for response time estimation (Baldwin et al., 2021). Models utilizing embeddings exhibited an average RMSE for the response time task exceeding that of models leveraging linguistic features by 12 seconds. This observation aligns with the intuitive notion that a greater word count in a question correlates with increased time required for student comprehension and analysis, consequently resulting in extended response times. The rationale lies in the fact that candidates typically need more time to read a question with a higher word count, thereby automatically increasing the response time.

6 Conclusion

In conclusion, our research demonstrates the efficacy of machine learning models and feature engineering in addressing key challenges of standardized testing. Linear Regression coupled with lin-

Serial number	Model	Featurization	RMSE for Task 1: Predicting Difficulty	RMSE for Task 2: Predicting Response Time
1.	One neuron network	Embeddings	0.368	32.708
2.	Fine-Tuned BERT	Embeddings	0.321	78.837
3.	Linear Regression	Embeddings	0.614	49.583
4.	Decision Trees	Embeddings	0.320	29.927
5.	KNN	Embeddings	0.332	29.727
6.	XGBoost	Embeddings	0.319	29.657
7.	GBDT	Embeddings	0.32	29.927
8.	Random Forest	Linguistic Features	0.303	26.234
9.	**Linear Regression**	**Linguistic Features**	**0.302**	**26.181**
10.	Decision Trees	Linguistic Features	0.348	28.862
11.	KNN	Linguistic Features	0.324	29.574
12.	XGBoost	Linguistic Features	0.353	28.644
13.	GBDT	Linguistic Features	0.348	28.862

Table 1: RMSE for different models and the corresponding featurization method

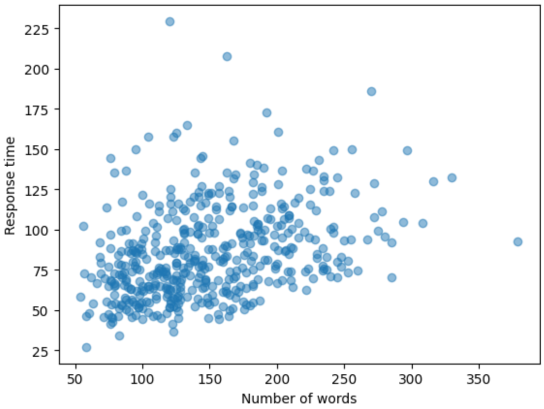


Figure 2: Number of words versus Response time

guistic features gave the lowest RMSE scores of 0.302 and 26.181 for the Difficulty prediction and response time prediction respectively. These findings highlight the potential of predictive models to streamline assessment processes and improve fairness. By reducing reliance on labor-intensive pretesting, our approach offers a scalable alternative while ensuring the integrity of assessment materials. Future research should explore additional techniques and validate findings across diverse educational contexts. Overall, our work advances educational assessment by offering innovative solutions to test design challenges.

References

Tahani Alsubait, Bijan Parsia, and Uli Sattler. 2013. A similarity-based theory of controlling mcq difficulty. pages 283–288.

Peter Baldwin, Victoria Yaneva, Janet Mee, Brian E. Clauser, and Le An Ha. 2021. Using natural language processing to predict item response times and improve test construction. *Journal of Educational Measurement*, 58(1):4–30.

Lisa Beinborn, Torsten Zesch, and Iryna Gurevych. 2015. Candidate evaluation strategies for improved difficulty prediction of language tests. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–11, Denver, Colorado. Association for Computational Linguistics.

Pritam Deka, Anna Jurek-Loughrey, and P Deepak. 2022. Improved methods to aid unsupervised evidence-based fact checking for online health news. *Journal of Data Intelligence*, 3(4):474–504.

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.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2020. Domain-specific language model pretraining for biomedical natural language processing.

Arya D. McCarthy, Kevin P. Yancey, Geoffrey T. LaFlair, Jesse Egbert, Manqian Liao, and Burr Settles. 2021. Jump-starting item parameters for adaptive language tests. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 883–899, Online and Punta Cana, Dominican Republic. 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.

Ulrike Padó. 2017. Question difficulty – how to estimate without norming, how to use for automated grading. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–10, Copenhagen, Denmark. Association for Computational Linguistics.

Burr Settles, Geoffrey T. LaFlair, and Masato Hagiwara. 2020. Machine Learning–Driven Language Assessment. *Transactions of the Association for Computational Linguistics*, 8:247–263.

Kang Xue, Victoria Yaneva, Christopher Runyon, and Peter Baldwin. 2020. Predicting the difficulty and response time of multiple choice questions using transfer learning. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 193–197, Seattle, WA, USA → Online. Association for Computational Linguistics.

Victoria Yaneva, Le An Ha, Peter Baldwin, and Janet Mee. 2020. Predicting item survival for multiple choice questions in a high-stakes medical exam. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6812–6818, Marseille, France. European Language Resources Association.

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.

Victoria Yaneva, Kai North, Peter Baldwin, An Ha Le, Saed Rezayi, Yiyun Zhou, Sagnik Ray Choudhury, Polina Harik, and Brian Clauser. 2024. Findings from the First Shared Task on Automated Prediction of Difficulty and Response Time for Multiple Choice Questions. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Leveraging Physical and Semantic Features of text item for Difficulty and Response Time Prediction of USMLE Questions

Gummuluri Venkata Ravi Ram[1], Kesanam Ashinee[2], Anand Kumar M[3]
Department of Information Technology, National Institute of Technology Karnataka
Surathkal, Karnataka, India
[1]toraviram2003@gmail.com, [2]ashineekesanam@gmail.com, [3]m_anandkumar@nitk.edu.in

Abstract

This paper presents our system developed for the Shared Task on Automated Prediction of Item Difficulty and Item Response Time for USMLE questions, organized by the Association for Computational Linguistics (ACL) Special Interest Group for building Educational Applications (BEA SIGEDU). The Shared Task, held as a workshop at the North American Chapter of the Association for Computational Linguistics (NAACL) 2024 conference, aimed to advance the state-of-the-art in predicting item characteristics directly from item text, with implications for the fairness and validity of standardized exams. We compared various methods ranging from BERT for regression to Random forest, Gradient Boosting(GB), Linear Regression, Support Vector Regressor (SVR), k-nearest neighbours (KNN) Regressor, Multi-Layer Perceptron(MLP) to custom-ANN using BioBERT and Word2Vec embeddings and provided inferences on which performed better. This paper also explains the importance of data augmentation to balance the data in order to get better results. We also proposed five hypotheses regarding factors impacting difficulty and response time for a question and also verified it thereby helping researchers to derive meaningful numerical attributes for accurate prediction. We achieved a RSME score of 0.315 for Difficulty prediction and 26.945 for Response Time.

1 Introduction

The automated prediction of item difficulty and item response time is a critical task in the field of educational assessment, with implications for the fairness and validity of standardized exams. Traditionally, item characteristics such as difficulty and response time have been obtained through labor-intensive pretesting processes, posing challenges related to time, cost, and security. To address these challenges, there is a growing interest in leveraging natural language processing (NLP) techniques to predict item characteristics directly from the item text. (Baldwin et al., 2021)

In this paper, we present our system developed for the Shared Task on Automated Prediction of Item Difficulty and Item Response Time, organized by the Association for Computational Linguistics (ACL) Special Interest Group for Building Educational Applications (BEA SIGEDU). The Shared Task was held as a workshop at the North American Chapter of the Association for Computational Linguistics (NAACL) 2024 conference. Our participation in this Shared Task aimed to advance the state-of-the-art in predicting item characteristics and contribute to the ongoing efforts to improve the efficiency and fairness of standardized testing.

In this paper, we provide an overview of our system architecture, including methodologies employed for predicting item difficulty and item response time. We describe the features utilized, the model architectures, and the training procedures. Furthermore, we present the experimental setup, including the dataset used for training and evaluation, data augmentation, as well as the evaluation metrics employed to assess the performance of our system as prescribed by shared task.

Through our participation in the Shared Task, we aim to demonstrate the effectiveness of our approach in predicting item characteristics and contribute to the collective efforts in developing more accurate and efficient models for automated assessment in educational contexts. Additionally, we discuss the implications of our findings and potential future directions for research in this area. We believe that our system holds promise for enhancing the fairness and effectiveness of standardized testing, ultimately benefiting both test developers and test takers alike.

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 534–541
June 20, 2024 ©2024 Association for Computational Linguistics

2 Related Work

The paper **"Automated Prediction of Item Difficulty in Reading Comprehension Using Long Short-Term Memory"** by Li-Huai Lin, Tao-Hsing Chang, Fu-Yuan Hsu focuses on utilizing Long Short-Term Memory (LSTM) to predict the difficulty of test items in reading comprehension. Traditional methods of estimating item difficulty relied on expert validation or pretests, which were labor-intensive and costly. By automating the prediction process using LSTM, the study aims to overcome these challenges. Experimental results indicate that the proposed method shows a good prediction agreement rate. The use of LSTM in predicting item difficulty offers a more efficient and accurate approach compared to manual methods, showcasing the potential of machine learning in educational assessment (Štěpánek et al., 2023)

The paper **Question Difficulty Prediction for READING Problems in Standard Tests** by Huang, Z., Liu, Q., Chen, E., Zhao, H., Gao, M., Wei, S., Su, Y. and Hu, G. initially involves converting the input into embeddings, followed by passing it through a Bidirectional Long Short-Term Memory (BI-LSTM) network to capture semantic relationships. Subsequently, an Attention Layer is employed to identify words within the document or option that hold significant relevance to a given question. This process aids in selecting pertinent information. Finally, the Prediction Layer displays the predicted difficulty scores. (Huang et al., 2017)

Jump-Starting Item Parameters for Adaptive Language Tests by the authors McCarthy, A.D., Yancey, K.P., LaFlair, G.T., Egbert, J., Liao, M. and Settles, B address the challenge of calibrating test item difficulties in high-stakes language assessments, either with limited pilot test data or without any prior information. They propose a multi-task generalized linear model utilizing BERT features to jump-start the estimation of item difficulties. With only 500 test-takers and a small sample of item exposures from a large item bank, their approach rapidly improves the quality of difficulty estimates. This joint model facilitates the comparison of test-taker proficiency, item difficulty, and language proficiency frameworks such as the Common European Framework of Reference (CEFR). Moreover, it allows for the generation of new item difficulty estimates without the need for piloting, reducing item exposure and enhancing test security. The authors validate their method using operational data from the Duolingo English Test, demonstrating strong correlations between the derived difficulty estimates and lexico-grammatical features associated with reading complexity. (McCarthy et al., 2021)

3 Task Description

In recent times, Efforts have been made to enhance the prediction of item parameters for high-stakes medical exams such as the USMLE, have been hampered by challenges in sharing exam data. To address this gap, A Shared Task is proposed focusing on predicting item parameters using practice item content and characteristics from the USMLE Examination. Refer (Yaneva et al., 2024)

The objective of this Shared Task is to advance the state-of-the-art in item parameter prediction, specifically focusing on two tracks:

- **Track 1 - Predicting Item Difficulty:** Given the item text and associated metadata, participants are tasked with predicting the item difficulty variable. Item difficulty represents the proportion of examinees who answer the item/question correctly, providing insights into the relative complexity of the item.

- **Track 2 - Predicting Item Response Time:** Given the item/question text and metadata, participants are challenged to predict the time intensity variable, reflecting the time required by examinees to respond to the item. Understanding item response time aids in optimizing exam administration and identifying potential issues with overly time-consuming items.

4 Dataset Description

The dataset for this Shared Task comprises 466 previously utilized and retired Multiple Choice Questions (MCQs) from the United States Medical Licensing Examination (USMLE) Steps 1, 2 CK, and 3. The USMLE is a series of examinations handled and developed by the National Board of Medical Examiners (NBME) and the Federation of State Medical Boards (FSMB) to support medical licensure decisions in the United States.

The dataset is structured with the following attributes:

ItemNum: Consecutive number assigned to the item in the dataset.

ItemStem_Text: Textual description of the clinical case or scenario presented in the MCQ stem.

Answer_A to Answer_J: Text for response options A to J. Unused columns remain blank for items with fewer than J response options.
Answer_Key: Letter denoting the correct answer for the item.
Answer_Text: Text corresponding to the correct response for the item.
ItemType: Denotes whether the item contained an image (PIX) or not (Text). Images are not part of the dataset.
EXAM: Indicates the Step of the USMLE exam to which the item belongs (Step 1, Step 2, or Step 3).
Difficulty: Measure of item difficulty where higher values indicate more difficult items.
Response_Time: Mean response time for the item measured in seconds, including initial response and revisits by examinees.

The guidelines for MCQ construction emphasize adherence to a standard structure, avoiding extraneous material, misleading information, and grammatical cues. The items were authored by experienced subject matter experts to assess medical knowledge.

The training data consists of 466 samples. Additonally, to augment the sample dataset, we employed paraphrasing on the provided textual questions (ItemStem_Text) and expanded the training dataset size.

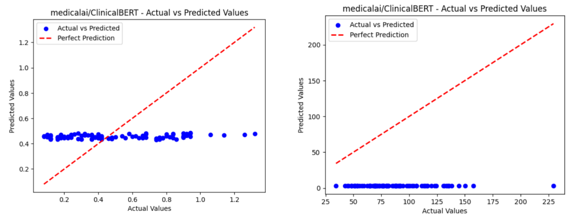
5 Methodology

5.1 Baseline Model

We tried BERT for regression as baseline model. We fine-tuned BERT specifically for regression tasks, utilizing BERT embeddings of the questions. Leveraged the CamembertTokenizer to process the textual descriptions from our dataset.

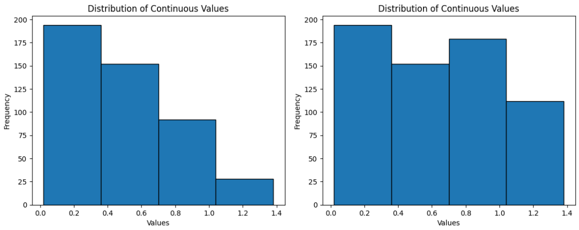
To ensure with BERT's maximum input sequence length of 512 tokens, we set a maximum input sequence length of 300 tokens. Any descriptions exceeding this length were filtered out to avoid truncation, ensuring the integrity of the input data.

The BERT architecture consists of an embedding layer and 12 stacked transformers. Each input sequence yields a sequence of vectors as output, with each vector representing a token in the input. However, for regression tasks, only the final hidden state of the first token, denoted by the "[CLS]" token, is utilized. In line with BERT's architecture, we appended a dense linear layer with dropout after the "[CLS]" token to serve as the final regression layer. This layer facilitates the regression task by mapping the BERT embeddings to the corresponding output labels.



(a) Difficulty (b) Response Time

Figure 1: Predicted v/s True Value plot on validation set on finetuning BERT as regressor



(a) Original Data (b) After augmntation

Figure 2: True Value Distribution in 4 bins before and after data augmentation

5.2 Data Augmentation

The training dataset provided comprises 466 samples. Upon analyzing the distribution of difficulty values, we observed a scarcity of samples with difficulty greater than 0.7. Consequently, we utilized GPT-3.5 LLM to generate additional instances through paraphrasing techniques. Passed on the question samples into the LLM and gave a prompt to paraphrase the given samples. Refer Fig.2a for imbalanced data and Fig.2b for balanced data

5.3 Data Engineering

We propose the following hypotheses based on literature review and reviews from students, based on experience:

- "The readability of a question influences its difficulty and response time" : The tougher the question is to read, the more the student gets confused and hence difficulty and average response time increases.
- "The average length of a question affects response time and subsequently, difficulty" : longer questions take long time to read.

- "The number of options may lead to confusion, potentially increasing difficulty"
- "The average length of options impacts response time and difficulty"
- "The similarity among options influences decision-making, thus affecting difficulty and response time"

Consequently, we extracted these features from the provided dataset. For readability assessment, we utilized the SMOG index (Lin et al., 2019), which is used in educational and medical settings to calculate readability of a document.

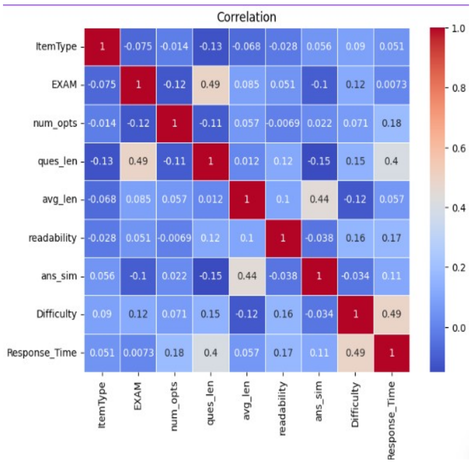


Figure 3: Correlation of extracted features with target variables

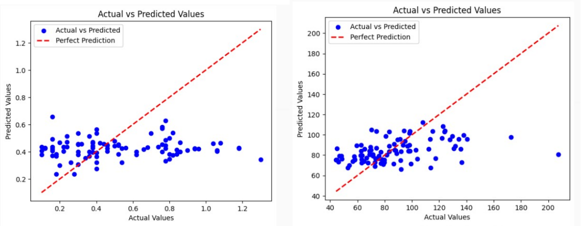
We can see the correlation heat-map as in Fig.3. Clearly the extracted features seem to have good correlation with difficulty and response time, thence justifying the hypotheses.

5.4 Bio-BERT Embeddings

The dataset, being from the medical domain, necessitated the utilization of BioBERT to extract embeddings. We fine-tuned BioBERT specifically based on question-difficulty pairs. The embeddings encapsulate contextual information aligned with the respective difficulty levels. (Yaneva et al., 2019) (Yaneva et al., 2020). In our exploration, we experimented with various methodologies and approaches

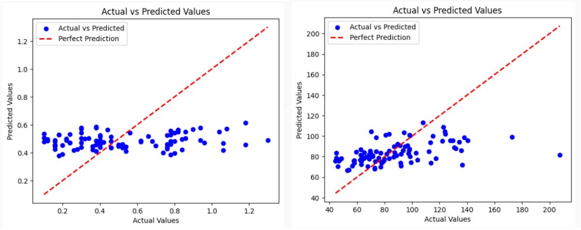
5.5 Approach I - BERT + ANN

We designed 2 distinct Artificial Neural Networks (ANNs) to explore the relationship between the features extracted from the dataset, particularly in the context of question difficulty.



(a) Difficulty (b) Response Time

Figure 4: Predicted v/s True Value plot on Val set for ANN 1 trained on Embeddings + Num Features



(a) Difficulty (b) Response Time

Figure 5: Predicted v/s True Value plot on Val Set for ANN 2 trained on embeddings + Num Features

For the first ANN architecture, we leveraged BioBERT embeddings, which are representations derived from a pre-trained language model specifically tailored for the biomedical domain. These embeddings, comprising vectors of size 768, served as one input stream. Concurrently, we processed seven numerical features independently. These features likely included various attributes such as question length, readability scores, and other relevant metrics. Each stream of inputs traversed through its respective hidden layers before being concatenated at a later stage, in order to capture intricate relationships between textual and numerical features.

The second ANN configuration adopted a different strategy. Here, we fused both the text embeddings obtained from BioBERT and the numerical feature vector derived from the dataset. By concatenating these representations, we aimed to create a unified feature set that encapsulates both textual and numerical attributes of the questions. This combined input was then fed through the hidden layers of the neural network, potentially enabling the model to discern intricate patterns and correlations between the textual content and numerical characteristics of the questions.

By employing these two distinct architectures, we aim to explore and compare the effectiveness of different approaches in utilizing BioBERT embeddings and numerical features to predict question difficulty levels within the medical domain.

Table 1: Results Of the 2 ANN Models

Target labels	**ANN1**	**ANN2**
Difficulty	0.32	0.29
Response Time	26.65	26.11

5.6 Approach 2 - Word2Vec + ML Models

The Deep learning models such as ANNs rely more on larger databases for optimal performance, we've opted for an alternative strategy. Hence we've transitioned to utilizing Word2Vec embeddings, a widely-used technique for generating word embeddings based on the distributional semantics of words within a corpus. Unlike BERT, which thrives on large datasets to capture contextual nuances, Word2Vec offers a computationally efficient means to represent words in a continuous vector space, thereby capturing semantic relationships.

For this, we trained regression models on the Word2Vec embeddings and specifically, we employed the following regression models:

1. Random Forest: An ensemble learning method capable of handling non-linear relationships and high-dimensional data, Random Forest constructs a multitude of decision trees during training and outputs the mean prediction of individual trees.

2. Linear Regression: A regression technique that models the relationship between the dependent variable and one or more independent variables by assuming a linear relationship between them.

3. Support Vector Regression (SVR): A regression algorithm based on the Support Vector Machine (SVM) framework, SVR is adept at handling non-linear relationships by mapping data into a higher-dimensional feature space.

By leveraging Word2Vec embeddings and training on these regression models, we aim to capture the intricate relationships between the textual representations of medical questions and their corresponding difficulty levels. (Yaneva et al., 2021)

Table 2: Word2Vec + ML Model (Linear Regression, SVR, Random Forest Regressor

Target Values	LR	SVR	RFR
Difficulty	0.37	0.356	0.324
Response Time	79.59	86.227	27.24

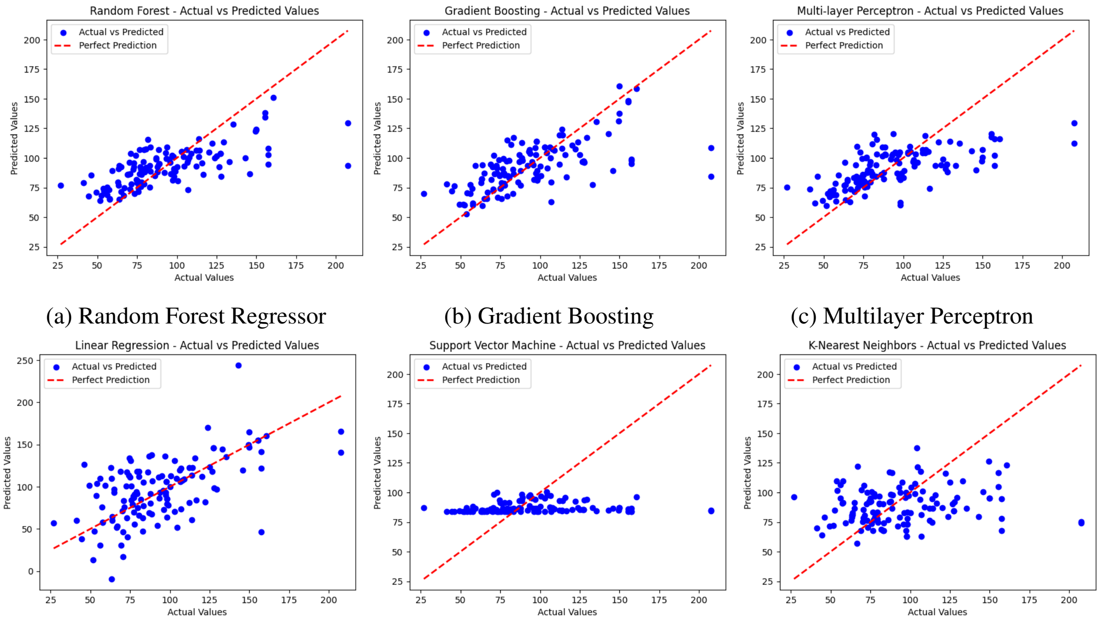
5.7 Approach 3 - BERT + ML Models

We performed experimentation utilizing BioBERT embeddings in three distinct configurations: only with text embeddings, only with numerical features, and with a concatenated dataset combining text embeddings and numerical features. The numerical features encompassed attributes such as average length, readability scores, number of options, average length of options, and similarity scores derived from the dataset. The concatenated dataset combines the text embeddings from BioBERT with the numerical features, aiming to leverage both the textual and quantitative aspects of the data for improved regression performance. Each of these datasets underwent training on a range of regression models, including Random Forest, Gradient Boosting, Linear Regression, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Multilayer Perceptron (MLP). By systematically exploring various feature combinations and regression algorithms, we aimed to discern the most effective methodologies for predicting the desired output labels.This comprehensive approach enables us to evaluate the performance of various models and feature combinations, thereby gaining insights into the most suitable methodologies for our regression task. (Settles et al., 2020)

6 Experimental Results and Discussion

Our baseline model, BERT Regressor, achieved RMSE scores of 0.307 for predicting Difficulty and 88.502 for predicting Response Time. These scores demonstrate the model's performance in predicting both the difficulty level of exam items and the time intensity required for examinees to respond to them. Fig.1a and Fig.1b shows that most of predicted values are in a specified range and hence we assumed that the imbalance in data as shown in Fig.2a. Hence we balanced the data. We also extracted few numerical features as discussed in feature engineering section and experimented with them.

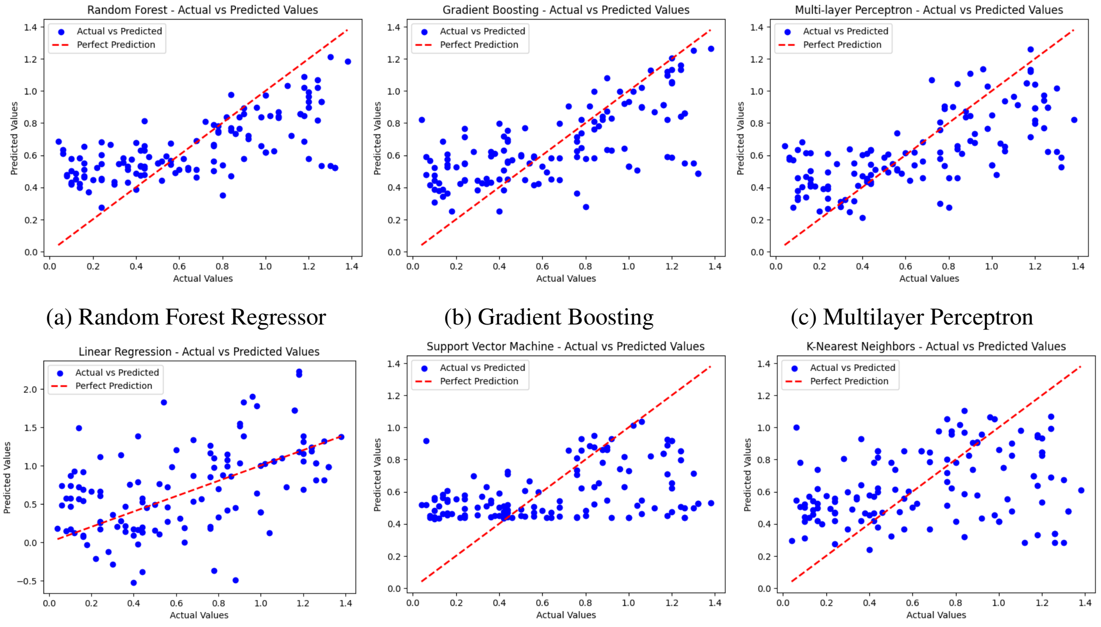
Instead of simply finetuning BERT, we trained Bio-BERT embeddings with ANN and results are as shown in table 1. We tried two ANNs whose architecture is as mentioned in methodology section, former concatenating numerical features and text embeddings in a hidden layer and the latter initially concatenating both. As shown in Fig.4a, Fig.4b and Fig.5a, Fig.5b the dispersion of predicted values increased but still not upto the mark. ANN1



(a) Random Forest Regressor (b) Gradient Boosting (c) Multilayer Perceptron

(d) Linear Regression (e) Support Vector Regressor (f) KNN Regressor

Figure 6: Plot of Predicted V/S true labels for validation dataset for **Difficulty** variable upon training ML models on concatenated Input (Text Embedding + numerical features)



(a) Random Forest Regressor (b) Gradient Boosting (c) Multilayer Perceptron

(d) Linear Regression (e) Support Vector Regressor (f) KNN Regressor

Figure 7: Plot of Predicted V/S true labels for validation dataset for **Response Time** variable upon training ML models on concatenated Input (Text Embedding + numerical features)

Target labels	**RFR**	**GB**	**LR**	**SVM**	**MLP**	**KNN**
Difficulty	0.294	0.283	0.480	0.296	0.329	0.293
Response Time	24.029	24.508	28301.258	31.959	25.363	26.918

Table 3: RMSE Scores for 6 models on two target labels: Difficulty and Response Time using only question embeddings

Target labels	RFR	GB	LR	SVM	MLP	KNN
Difficulty	0.346	0.331	0.370	0.363	0.390	0.417
Response Time	28.86	29.37	32.28	32.24	33.21	34.47

Table 4: RMSE Values across 6 models on two target labels: Difficulty and Response Time using only numerical data

Target labels	RFR	GB	LR	SVM	MLP	KNN
Difficulty	0.292	0.297	0.489	0.362	0.39	0.288
Response Time	24.05	24.47	31.48	32.27	33.38	25.05

Table 5: RMSE Values accross 6 models on two target labels: Difficulty and Response Time using concatenated data

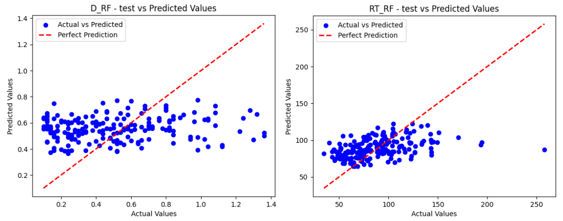
performed better than ANN2 in increasing range of prediction. Hence we assumed it might be due to BERT being a Large Language Model is unable to capture the essence or overall context with such small dataset, and hence shifted to more general model Word2Vec with ML Models as we presume DL models need more data.

Moving on to our 2nd approach consisting of training ML models with Word2Vec embeddings, the results are as in Table 2 . Clearly results are worser when compared to that of training BERT embeddings with ANN.

Hence we considered the issue is in ANN. Since ANN being Deep Learning Model, with such limited data it is unable to capture patterns essentially and hence we tried training ML models with Bio-BERT embeddings. They outperformed ANN, hence we came to conclusion of using ML models for prediction.

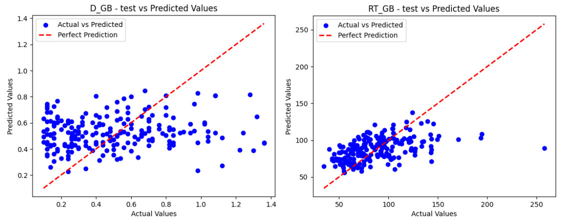
In order to understand importance of extracted numerical features, we used the same ML models to perform regression on only question embeddings and only numerical data and results for each are shown in Table 3 and Table 4 respectively. This clearly states that both text-embeddings and numerical features engineered by our hypotheses are crucial for predicting values.

Hence we concatenated both and trained the ML models to get results as shown in Table 5. Clearly Gradient Boosting, Random Forest Regressor and Multi Layer Perceptron have performed best and hence we considered them to be best models for submission. Fig. 6a - Fig. 6f shows the plots for actual v/s predicted Diffculty values. Fig. 7a - Fig. 7f shows the plots for actual v/s predicted Response Time values



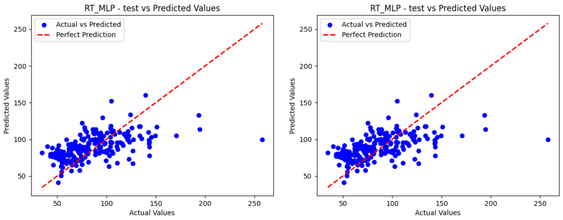
(a) Difficulty (b) Response Time

Figure 8: Actual v/s Predicted value plots for Random Forest Regressor on gold_label test data



(a) Difficulty (b) Response Time

Figure 9: Actual v/s Predicted value plots for Gradient Boosting on gold_label test data



(a) Difficulty (b) Response Time

Figure 10: Actual v/s Predicted value plots for Multi-Layer Perceptron on gold_label test data

Table 6: Test Data Results

Target labels	RFR	GB	MLP
Difficulty	0.315	0.322	0.336
Response Time	28.768	27.481	26.945

7 Error Analysis

The final test data for shared task had 201 data points and Team ScalarLab had made three submissions/prediction files obtained by best three models of which we trained viz. Random Forest Regressor, Gradient Boosting and Multilayer Perceptron trained on concatenated Bio-BERT embeddings and extracted numerical features. The RMSE scores are as reported in Table6. Fig.8, Fig.9 and Fig.10, we clearly can see they outperformed ANN and BERT for regressor.

8 Conclusion and Future Work

We have achieved 0.315 RMSE for difficulty prediction and 26.945 RMSE for response time prediction. We successfully compared and explained why Deep Learning model ANN failed in making better predictions, we discussed the importance of data augmentation and how results improved, and also proposed five hypotheses that seem to impact difficulty, response time of MCQs. As future work, we would like to explore how Deep Learning Models can learn better with limited data and which embeddings are better fir such tasks where limited data is available. We would also explore what are ther factors that impact difficulty and response time of questions (MCQs) and incorporate that info in models to be trained to achieve better RMSE scores.

References

Peter Baldwin, Victoria Yaneva, Janet Mee, Brian E Clauser, and Le An Ha. 2021. Using natural language processing to predict item response times and improve test construction. *Journal of Educational Measurement*, 58(1):4–30.

Zhenya Huang, Qi Liu, Enhong Chen, Hongke Zhao, Mingyong Gao, Si Wei, Yu Su, and Guoping Hu. 2017. Question difficulty prediction for reading problems in standard tests. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Li-Huai Lin, Tao-Hsing Chang, and Fu-Yuan Hsu. 2019. Automated prediction of item difficulty in reading comprehension using long short-term memory. In *2019 international conference on asian language processing (ialp)*, pages 132–135. IEEE.

Arya D McCarthy, Kevin P Yancey, Geoffrey T LaFlair, Jesse Egbert, Manqian Liao, and Burr Settles. 2021. Jump-starting item parameters for adaptive language tests. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 883–899.

Burr Settles, Geoffrey T. LaFlair, and Masato Hagiwara. 2020. Machine learning–driven language assessment. *Transactions of the Association for computational Linguistics*, 8:247–263.

Lubomír Štěpánek, Jana Dlouhá, and Patrícia Martinková. 2023. Item difficulty prediction using item text features: Comparison of predictive performance across machine-learning algorithms. *Mathematics*, 11(19):4104.

Victoria Yaneva, Peter Baldwin, Janet Mee, et al. 2019. Predicting the difficulty of multiple choice questions in a high-stakes medical exam. In *Proceedings of the fourteenth workshop on innovative use of NLP for building educational applications*, pages 11–20.

Victoria Yaneva, Peter Baldwin, Janet Mee, et al. 2020. Predicting item survival for multiple choice questions in a high-stakes medical exam. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6812–6818.

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. Association for Computational Linguistics.

Victoria Yaneva, Kai North, Peter Baldwin, An Ha Le, Saed Rezayi, Yiyun Zhou, Sagnik Ray Choudhury, Polina Harik, and Brian Clauser. 2024. Findings from the First Shared Task on Automated Prediction of Difficulty and Response Time for Multiple Choice Questions. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

UPN-ICC at BEA 2024 Shared Task: Leveraging LLMs for Multiple-Choice Questions Difficulty Prediction

George Dueñas[1], **Sergio Jimenez**[2], **Geral Eduardo Mateus Ferro**[3]
[1]Doctorado Interinstitucional en Educación, Universidad Pedagógica Nacional, Colombia
[2]Instituto Caro y Cuervo, Colombia
[3]Departamento de Lenguas, Universidad Pedagógica Nacional, Colombia
geduenasl@upn.edu.co, sergio.jimenez@caroycuervo.gov.co, gmateus@pedagogica.edu.co

Abstract

We describe the second-best run for the shared task on predicting the difficulty of Multi-Choice Questions (MCQs) in the medical domain. Our approach leverages prompting Large Language Models (LLMs). Rather than straightforwardly querying difficulty, we simulate medical candidate's responses to questions across various scenarios. For this, more than 10,000 prompts were required for the 467 training questions and the 200 test questions. From the answers to these prompts, we extracted a set of features which we combined with a Ridge Regression to which we only adjusted the regularization parameter using the training set. Our motivation stems from the belief that MCQ difficulty is influenced more by the respondent population than by item-specific content features. We conclude that the approach is promising and has the potential to improve other item-based systems on this task, which turned out to be extremely challenging and has ample room for future improvement.

1 Introduction

The item difficulty is a core problem in the construction of exams. The exam items should encompass a broad spectrum of difficulty levels to efficiently ascertain the competencies of the test takers being assessed. Traditionally, item difficulty has been a manual task done by human experts (Lorge and Diamond, 1954; Haladyna et al., 2002) despite its inherent disadvantages compared to other approaches based on data (Wauters et al., 2012; Choi and Moon, 2020). Nevertheless, recent progress in Natural Language Processing (NLP) has facilitated the automated prediction of item difficulty from textual content (Dueñas et al., 2015; Benedetto, 2023), serving as an alternative to traditional pretesting and manual task (AlKhuzaey et al., 2023; Benedetto et al., 2023).

These recent studies underscore the growing importance and interest in the topic of item difficulty prediction. In response, BEA has launched the Shared Task "Automated Prediction of Item Difficulty and Item Response Time" (Yaneva et al., 2024). This initiative represents an effort to push the boundaries of current research in item parameter prediction. The data provided for this task includes multiple-choice questions from Steps 1, 2 CK, and 3 of the USMLE, which is a sequence of examinations used to facilitate medical licensing in the United States.

Recent studies have leveraged NLP and Machine Learning techniques to address these challenges, providing insight into the factors that contribute to difficulty of Multiple-Choice Questions (MCQs). Four seminal studies are reviewed below that, together, show the approaches and advances that have been made in the automated prediction of USMLE item difficulty.

Ha et al. (2019) laid foundational work by developing a method to estimate the difficulty of USMLE MCQs based on a diverse array of linguistic features and embedding types (ELMo and Word2Vec), including measures quantifying the difficulty for an automated question-answering system. Their approach surpassed various baselines significantly (ZeroR, Word Count, Average Sentence Length, Average Word Length in Syllables, and the Flesch Reading Ease formula). The study emphasized that information from all levels of linguistic processing contributes to item difficulty, with semantic ambiguity and psycholinguistic properties of words being particularly influential.

In an study by Yaneva et al. (2020), they provide an approach towards predicting item survival using linguistic features, two types of embeddings (Word2Vec and ELMo), and Information Retrieval (IR) features in a high-stakes medical exam context. They implemented these features within a Random Forests algorithm framework and validated their approach using a dataset of 5,918 pretested MCQs from USMLE. Their findings indicated that the

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 542–550
June 20, 2024 ©2024 Association for Computational Linguistics

combination of all feature types outperformed the baselines, with ELMo being the strongest individual predictor, followed by Word2Vec, linguistic features, and IR features.

Xue et al. (2020) explored the application of transfer learning to predict the item difficulty and response time for approximately 18,000 MCQs from USMLE. They used three types of item text configurations as input: i) item stem, ii) item options, and iii) a combination of the stem and options. They were used to train three different ELMo models. This research demonstrated that while transfer learning significantly enhances predictions for response time, when item difficulty is used as an auxiliary task, the converse is not true. Difficulty prediction was most effective using signals from the item stem, while response time was best predicted using information from the entire item.

Building on Ha et al. (2019) approach, Yaneva et al. (2021) classified 18,961 MCQs from Step 2 of the USMLE into two categories in an unsupervised way: low-complexity items and high-complexity items, with the purpose of identifying interpretable relationships between item text and item complexity. They maintain that examining the linguistic features of the items can assist test developers in gaining a more detailed understanding of how cognitively more complex items differ from those with more straightforward solutions. Similar to previous studies, they provide empirical evidence that linguistic features, both syntactic and semantic, play a crucial role in determining the complexity associated with the item response process.

Unlike previous studies, we investigated the hypothesis that item difficulty depends more on the features of the test-taking population than on the items themselves. To explore this, we simulated medical students' answers to various MCQs across different examinations by prompting a Large Language Model (LLM). This approach allowed us to understand how certain features influence item difficulty, providing insights that challenge previous methods of educational assessment. In this paper we describe our participating system in the BEA 2024 Shared Task: Automated Prediction of Item Difficulty, which used a LLM as core approach.

2 System Description

2.1 Data

The data consist of a collection of 667 MCQs from USMLE Steps 1, 2 CK, and 3, which were used and now are retired (467 for training and 200 for test). These items have the traditional information, which is composed of a case (stem), the correct answer (key), the incorrect answer options (distractors), and the answer text, which contains the text of the correct response for the item. Moreover, each item comes with supplemental details as follows: item type, where "Text" indicates items composed entirely of text without images, while "PIX" represents items that include images, but these are not part of the dataset; EXAM specifies the Step of the USMLE exam the item belongs to (Step 1, Step 2, or Step 3); item difficulty, where higher values indicate more difficult items, and time intensity, which is the arithmetic mean response time, measured in seconds, across all examinees who attempted a given item in a live exam.

2.2 Features extracted from the items

The task consist of predicting automatically the item difficulty using approximately the 70% of items as training and the other part as test bed. Our approach consists in extracting 4 different sets of features from answers of ChatGPT-3.5 to different prompts, and a regression algorithm for predicting the ground truth labels in the test set.

2.2.1 Features from LLM answering the questions

This first set of features has been extracted from the process of asking the LLM to answer MCQs. The prompt used for this purpose is described below:

PROMPT #1

```
{Item_Stem_Text}
A: {Answer__A}
B: {Answer__B}
...

First, answer the question by providing
    only the letter of the option.
    Second, provide a brief explanation
    of your choice, but do not discuss
    other options or alternative
    scenarios.
```

Here, `{Item_Stem_Text}` is the text of the item, encompassing a comprehensive explanation of the medical case. The last sentence of the explanation is the question to be answered (e.g. "Which of the following is the most likely nutritional deficiency?"). Moreover, `{Answer__X}` denotes the textual content corresponding to each of the alternative option (e.g. "Vitamin D"). The context of the role in the completion chat for GPT-3.5 was: "Your are a medical doctor".

The main motivation for this prompt is to determine whether or not the LLM is capable of answering the questions. In principle, if the LLM is unable to answer correctly, this is an indication that the question is of high difficulty, and the opposite is also true. Additionally, we asked the LLM to provide a justification for its response to the prompt[1], from which we assume that extensive explanations are associated with high-difficulty questions and the opposite. Finally, in this group of features, we include some basic information about the item such as the length of distractors, the length of the correct option, among others, as indicators of the item difficulty. Below we detail the extracted features:

INCORRECT: Boolean indicating whether or not the question was answered correctly by the LLM.

JUSTIFICATION: Number of characters in the LLM's answer after removing the text of the option selected.

DISTRACTORS: Length in characters of the LLM response minus the length of the correct option text.

STEM: Length in characters of `Item Stem Text`.

KEY: Length in characters of the correct option.

STEM/KEY: The ratio between STEM and KEY features.

GPT_RESPONSE_TIME: Time in milliseconds reported by the LLM to answer the question.

COMPLETION_TOKENS: Number of tokens in the response reported by the LLM.

PROMPT_TOKENS: Number of tokens in the prompt reported by the LLM.

EXAM: Metadata of the item obtained from the dataset denoting the Step of the USMLE exam the item belongs to (Step 1, Step 2, or Step 3).

2.2.2 Features from splitting the items into yes/no questions

Given that the set of features from the previous subsection provides in the feature INCORRECT only a Boolean indication of the item difficulty, we employ the strategy of generating for each item a YES/NO sub-item for each option available in the item. In this way, the correctness of the LLM responses to these extracted sub-items provides more detailed indications of the difficulty of the original item. In this scenario, only one of the sub-items has the answer YES, and NO for the others. For this, we use the following prompt:

PROMPT #2:

```
{Item_Stem_Text}

First, answer clearly YES or NOT if
    Answer X is the correct answer to
    the question. Second, provide a
    brief explanation of your answer,
    but do not discuss other options.
```

Thus, if a question has n answer options, we generate n prompts for the LLM, from whose answers we extract the following features for each item:

YN_INCORRECT: Number of sub-items answered correctly for the item.

YN_INCORRECT_KEY: Boolean indicating whether the sub-item corresponding to the correct option was answered correctly or not by the LLM.

YN_OPTION_COUNT: Total number n of answer sub-items (options) for the item.

YN_YES_ANSWERS: Number of sub-items to which the LLM responded affirmatively.

YN_RESPONSE_TIME: Sum of the answer times for all sub-items reported by the LLM.

YN_JUSTIFICATION_CHAR: Sum of the lengths of the justifications (in characters) for the answers provided by the LLM to each sub-item.

YN_JUSTIFICATION_CORRECT: Sum of the lengths of the justifications for the answers to the sub-items that the LLM answered correctly.

YN_JUSTIFICATION_INCORRECT: Sum of the lengths of the justifications for the answers to the sub-items that the LLM answered incorrectly.

YN_JUSTIFICATION_YES: Sum of the lengths of the justifications for the answers to the sub-items that the LLM answered affirmatively.

[1] In our experiments, the LLM did not refuse to answer any questions, and thus it never stated that it is unable to provide information as an AI language model.

YN_JUSTIFICATION_NOT: Sum of the lengths of the justifications for the answers to the sub-items that the LLM answered negatively.

YN_JUSTIFICATION_KEY: Length of the justification for the sub-item corresponding to the correct option of the item.

YN_JUSTIFICATION_OPTIONS: Sum of the lengths of the justifications for the sub-items whose answer is NO.

YN_YES_OPTIONS Total number of affirmative answers given by the LLM to the sub-items.

YN_NOT_OPTIONS: Total number of negative answers given by the LLM to the sub-items.

YN_ALL_YES: Boolean indicating whether all answers to the sub-items were affirmative.

YN_ALL_NOT: Boolean indicating whether all answers to the sub-items were negative.

2.2.3 Features from using "mutilated" stems

In the features described in the previous subsections, the LLM has played a role equivalent to a test taker who has read all the texts of the questions and the options in detail. However, in real-life situations this is not always the case, and test takers have time pressures or personal preferences in reading with different "skimming" or "scanning" processes, which lead them to voluntarily or involuntarily omit some words while reading.

Our assumption is that highly-difficult items should be read in detail so that they can be answered correctly. On the contrary, in low-difficulty items, some words of their content can be omitted without this affecting their difficulty. To simulate this situation, we generate different modified versions of each item by incrementally "mutilating" the *stem*, randomly removing a percentage of its content words.

For this, we first tokenize sentences and identify which words or tokens in the *stem* are content words, marking the stopwords[2], which we exclude from the "mutilation" process. Likewise, we leave the last sentence of the *stem* intact, which contains the specific question of the item. Then, we set a percentage p, say $p = 0.20$, and randomly remove 20% of the content tokens from the *stem* (i.e. no stopwords). In this way, an item that remains answerable after a certain degree of mutilation of the stem would be an indicator of its level of difficulty. For this, we use a prompt similar to Prompt 1, but we mutilate the stem of each item at different percentages:

PROMPT #3:

```
{Item_Stem_Mutilated(P)}
A: {Answer__A}
B: {Answer__B}
...

First, answer the question providing
    only the letter of the option.
    Second, provide a brief explanation
    of your choice, but do not discuss
    other alternative options or
    scenarios.
```

Here P represents the percentage of mutilation of the stem. For each ítem, we used eight percentages ranging from 10%, 20%, 30%, until 80%. The following set of features is motivated by the assumption described above. Below we detail the extracted features:

MUT_10_INCORRECT: Boolean indicating if the LLM answered correctly the question in spite of the stem being mutilated at 10% of its content words.

MUT_20_INCORRECT: *Idem* Boolean indicating if the LLM answered correctly the question in spite of the stem being mutilated at 20% of its content words (other six features for 30%, 40%, 50%, 60%, 70%, and 80%.

MUT_INCORRECT: Number of incorrect answers out of the 8 levels of percentage of mutilation.

FIRST_MUT_INCORRECT: The lowest percentage of mutilation in which the LLM failed to answer the question correctly. If feature INCORRECT value is true, then this feature is zero.

LAST_MUT_INCORRECT: The highest percentage of mutilation where the LLM failed to answer the question correctly. If feature INCORRECT value is true, then this feature is zero.

FIRST_MUT_CORRECT: The lowest percentage of mutilation where the LLM failed to

[2] We use sentence tokenizer and the list of stopwords for English in the Natural Language Toolkit https://www.nltk.org/search.html?q=stopwords

answer the question correctly. If feature INCORRECT value is false, then this feature is zero.

LAST_MUT_CORRECT: The highest percentage of mutilation where the LLM failed to answer the question correctly. If feature INCORRECT value is false, then this feature is zero.

2.2.4 Features from modified "temperatures"

"Temperature" is a parameter in ChatGPT that controls the level of randomness or "creativity" in the answers of this LLM. In the features described in the previous subsections, this parameter was set at $t = 1.0$, which is its default value that indicates an intermediate value between the extremes $p = 2.0$ (maximum randomness) and $p = 0.0$ (fully deterministic). By varying this parameter, it is possible to simulate different test takers with a single LLM.

In principle, we assume that test takers with low temperature are capable of objectively answering questions of all levels of difficulty. As the temperature gradually increases, the simulated test taker begins to reduce their objectivity and begins to be unable to correctly answer high-difficulty questions. In this way, if an item is only answered correctly by test takers with low temperature, then this is an indication of high difficulty in the item. Similarly, items that are answered correctly despite the high temperature of the test takers should indicate a low level of difficulty.

To extract features using this idea, we use Prompt #1 by varying the parameter t in the ChatGPT API call. We use 11 values of t, starting at $t = 0.0$ and increasing in increments of 0.2 up to $t = 2.0$. The following is the set of features obtained with this strategy:

TEMP_0.0_INCORRECT: Boolean indicating whether the LLM answered incorrectly the item using $t = 0$.

TEMP_0.2_INCORRECT: Boolean indicating whether the LLM answered incorrectly the item using $t = 0.2$.

TEMP_0.4_INCORRECT: Boolean indicating whether the LLM answered incorrectly the item using $t = 0.4$ and six other features that range from t to 2.0 ($t = 1.0$ was omitted because is identical to the feature INCORRECT).

TEMP_INCORRECT: Number of incorrect answers for the ítem out of the 11 values of t used.

FIRST_TEMP_INCORRECT: The lowest value of t where the LLM answered the question incorrectly.

LAST_TEMP_INCORRECT: The highest value of t where the LLM answered the question incorrectly.

AVG_TEMP_INCORRECT: Feature TEMP_INCORRECT divided by 11 (i.e. the number of used values for t).

FIRST_TEMP_CORRECT: The lowest value of t where the LLM answered the question correctly.

LAST_TEMP_CORRECT: The highest value of t where the LLM answered the question correctly.

AVG_TEMP_CORRECT: Number of correct answers for the item of the 11 values of t used divided by 11.

2.3 Experimental Setup

The official performance metric for the shared task is the Root-Mean Squared Error (RMSE) between the known difficulty levels of the items and the predictions made by the automatic system being evaluated. To use this metric in the evaluation of individual features, we fit a simple linear regressor, taking the feature as the independent variable and the known difficulty levels as the dependent variable. Since in this specific task the RMSE metric shows little variance between the different features, we propose the Spearman's rank correlation coefficient as an alternative measure.

Unlike RMSE, Spearman's correlation not only indicates whether the feature is positively or negatively correlated, but also provides the level of statistical significance (p-value). Therefore, under these two measures, a desirable feature will show low values of RMSE and high absolute values in Spearman's correlation. The predictive model used to combine the features with the training data was a Ridge regression, in which the regularization parameter α was adjusted with the aim of selecting a reduced the number of relevant features in the model. To evaluate this model, the training data was divided into 30 random partitions, assigning

90% of the data for training and 10% for testing in each partition. Subsequently, the RMSE measure was calculated for each of the 30 test partitions and the average of these results was reported.

3 Results

3.1 Feature performance

Table 1 shows the RMSE rates and Spearman's correlation for the features derived from the use of Prompt #1. In this group, only the INCORRECT, STEM, and KEY features produced significant correlations. Among them, KEY was the only feature that produced a negative correlation.

Feature	RMSE	Spearman
INCORRECT	0.298	0.259††
STEM	0.304	0.118†
DISTRACTORS	0.305	-0.085
KEY	0.306	-0.108†
EXAM	0.306	0.089
PROMPT_TOKENS	0.306	0.082
STEM/KEY	0.307	0.163††
JUSTIFICATION	0.308	0.028
GPT_RESPONSE_TIME	0.308	0.023
COMPLETION_TOKENS	0.308	0.019

††: $p < 0.01$; †: $p < 0.05$

Table 1: Performance of the features extracted from Prompt #1

Table 2 shows the same types of results for the features extracted from the use of Prompt #2. Unlike the results presented in Table 1, RMSE and Spearman measures show high agreement.

Table 3 shows the RMSE rates and correlations obtained from the prompts that incrementally mutilated the words in the items' stem. All of these features produced highly significant results. As anticipated based on our motivations, the FIRST_MUT_INCORRECT feature exhibited a strong negative correlation. This correlation suggests that if the LLM can still answer effectively to a highly distorted question, it serves as evidence of the low-difficulty item.

Figure 1 presents the relationship between the percentage of correct answers of the LLM and the variation of the percentage of stem multilation. The bars indicate a trend where the percentage of correct answers declines as the level of stem mutilation increases.

Feature	RMSE	Spearman
YN_JUSTIFICATION_CHAR	0.304	0.134††
YN_JUSTIFICATION_OPTIONS	0.304	0.122††
YN_JUSTIFICATION_INCORRECT	0.305	0.152††
YN_INCORRECT_KEY	0.305	0.145††
YN_RESPONSE_TIME	0.305	0.118†
YN_INCORRECT	0.306	0.131††
YN_JUSTIFICATION_NOT	0.306	0.087
YN_JUSTIFICATION_KEY	0.307	0.100†
YN_OPTION_COUNT	0.307	0.100†
YN_ALL_NOT	0.307	0.074
YN_YES_OPTIONS	0.307	0.065
YN_JUSTIFICATION_YES	0.308	0.022
YN_JUSTIFICATION_CORRECT	0.308	0.010
YN_YES_ANSWERS	0.308	-0.008
YN_NOT_OPTIONS	0.308	-0.007
YN_ALL_YES	0.308	0.022

††: $p < 0.01$; †: $p < 0.05$

Table 2: Performance of the features extracted from Prompt #2 by using the strategy of dividing the item into yes/no sub items.

Feature	RMSE	Spearman
FIRST_MUT_INCORRECT	0.300	-0.260††
MUT_INCORRECT	0.301	0.234††
LAST_MUT_INCORRECT	0.302	0.269††
LAST_MUT_CORRECT	0.303	-0.247††
INCORRECT_MUT_40	0.303	0.207††
INCORRECT_MUT_10	0.303	0.198††
INCORRECT_MUT_20	0.303	0.195††
FIRST_MUT_CORRECT	0.304	0.247††
INCORRECT_MUT_70	0.304	0.183††
INCORRECT_MUT_50	0.305	0.148††
INCORRECT_MUT_80	0.305	0.147††
INCORRECT_MUT_60	0.306	0.142††
INCORRECT_MUT_30	0.307	0.108†

††: $p < 0.01$; †: $p < 0.05$

Table 3: Performance of the features extracted from the usage of the strategy of randomly mutilating words from stems

Table 4 shows the results of Prompt #1 varying the parameter t (temperature) of the LLM. This set of features produced the best results for both performance measures. In particular, the best feature is FIRST_TEMP_INCORRECT, which obtained a negative correlation as expected by our motivations.

Figure 2 presents that increasing the temperature t reduces the LLM's ability to answer items correctly. Therefore, if the LLM set to a high temperature can still answer an item correctly, this reveals a low-difficulty item.

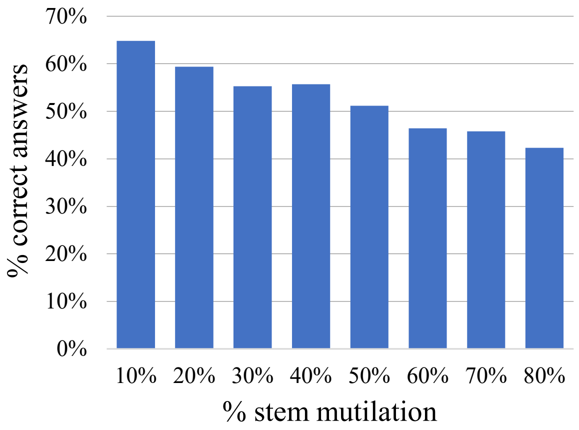


Figure 1: Percentage of correct answers in training data as stem mutilation varies.

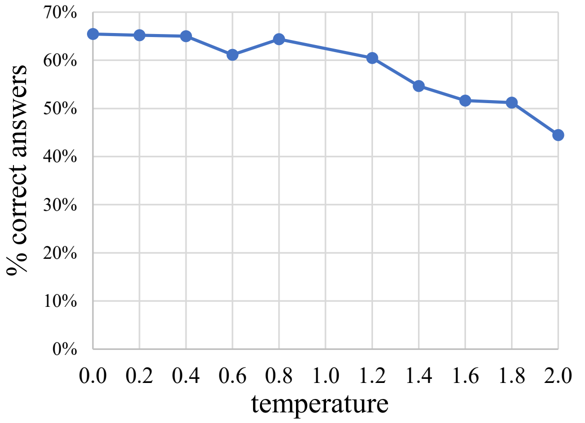


Figure 2: Percentage of correct answers in training data as the LLM temperature parameter varies.

Feature	RMSE	Spearman
FIRST_TEMP_INCORRECT	0.296	-0.293††
TEMP_INCORRECT	0.296	0.287††
TEMP_0.4_INCORRECT	0.297	0.261††
TEMP_0.2_INCORRECT	0.297	0.267††
TEMP_0.0_INCORRECT	0.298	0.254††
FIRST_TEMP_CORRECT	0.300	0.254††
TEMP_1.2_INCORRECT	0.300	0.236††
TEMP_1.6_INCORRECT	0.300	0.244††
TEMP_0.6_INCORRECT	0.301	0.232††
TEMP_0.8_INCORRECT	0.301	0.221††
LAST_TEMP_CORRECT	0.302	-0.181††
TEMP_1.4_INCORRECT	0.303	0.191††
LAST_TEMP_INCORRECT	0.303	0.17††
TEMP_1.8_INCORRECT	0.305	0.165††
TEMP_2.0_INCORRECT	0.305	0.131††

††: $p < 0.01$; †: $p < 0.05$

Table 4: Performance of the features extracted from varying temperature parameter in LLM.

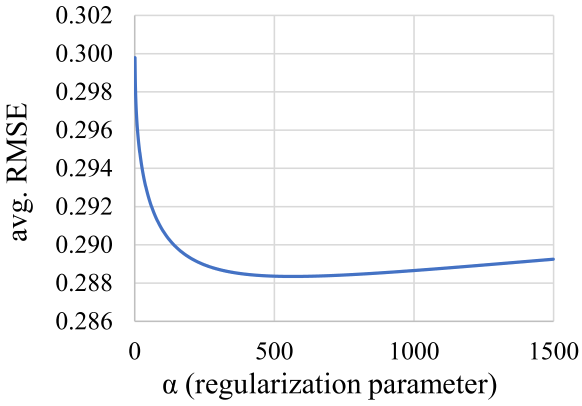


Figure 3: Performance in the training dataset of the item-difficulty prediction system as the regularization parameter α varies.

Finally, Figure 3 shows the results of the predictive system, which combines all the features based on the regularization parameter α of the Ridge Regression. As α increases, the RMSE rate decreases rapidly until it reaches the interval $500 < \alpha < 1000$, where a minimum is reached at $\alpha = 756$, which was the value of the parameter used for the final predictive model.

3.2 Submitted Run Results

This system generated predictions by extracting the previously described features from all items in the dataset. Next, a Ridge regression model was trained using the designated dataset, as this regression provided the best balance between performance and interpretability. This model produced the predictions for the test part of the dataset.

The official result obtained by our system (identified by the prefix UPN-ICC) is shown in Table 5, along with those obtained by other 4 top-performing systems out of 43 participating systems. Our single run produced notably competitive results, ranking 2nd in the task of predicting item difficulty. However, the best results barely surpassed the DummyRegressor baseline by a minimal margin, indicating that this task remains challenging.

4 Discussion

The results presented in Table 1 indicate that the INCORRECT feature emerges as the most significant predictor derived from the answers to Prompt

Team Name	Run	RMSE
EduTec	electra	0.299
UPN-ICC	run1	**0.303**
EduTec	roberta	0.304
ITEC	RandomForest	0.305
BC	ENSEMBLE	0.305
Baseline	DummyRegressor	0.311

Table 5: Results for task. The team name UPN-ICC is the system described in this document.

1. This feature is not directly derived from the item, but rather from the result obtained after exposing said item to a test taker, in this context simulated by the LLM. This finding supports our initial hypothesis, suggesting that an LLM can adequately simulate a test taker human behavior when facing the challenge of responding to MCQ items. However, contrary to our initial expectations, the lengths of the explanations provided by the LLM (JUSTIFICATION feature) did not prove to be predictive of the item difficulty.

Regarding the strategy of decomposing the MCQ item into YES/NO questions, as presented in Table 2, the results suggest that the YN_INCORRECT feature did not provide any additional significant information to improve the understanding provided by the INCORRECT feature, which constituted our main motivation for exploring this set of features. Nonetheless, the length of the justifications provided by the LLM to the YES/NO questions, in the YN_JUSTIFICATION_CHAR, _OPTIONS, and _INCORRECT features, resulted in a significant improvement in the performance of the JUSTIFICATION feature. This suggests that the strategy of decomposing the item into sub-items is effective, as it provides detailed justifications for each option of the MCQs, which are reliable indicators for predicting of item difficulty.

The results from Table 3 and Figure 1 indicate that the strategy of mutilating the stem text of the items to different degrees produces good predictors of item difficulty. This is an indication that this strategy allows for the simulation of different test takers with varying reading strategies using a single LLM. Furthermore, the analysis of the results presented in Table 3 reveals that the performance measure RMSE does not indicate significant differences among the features evaluated in this group. On the other hand, the Spearman correlation coefficient provides insightful results.

Similarly to the mutilation strategy, variations applied to the temperature parameter t resulted in efficient predictors of item difficulty (Table 4). It is noteworthy that, within the total training set, the percentage of correct responses ranges between 65% to 43% when varying both mutilation and temperature. This suggests that these two distinct strategies effectively simulate various types of test takers.

Since item difficulty is determined from item answers by a heterogeneous human population, the implementation of strategies to simulate this population is important in the effort to predict item difficulty. Given that these two strategies produced the most effective predictors in our system, exploring combinations of these and other similar strategies emerges as a promising research perspective for addressing this challenging task.

Finally, Figure 3 shows that the single regression system parameter, α, exhibits robust behavior over a wide range of its values, which likely contributed to the good performance of our system in the task.

5 Conclusion

We conclude that the strategy of simulating test takers using LLMs offers a novel and promising perspective for the prediction of MCQ difficulty. The strategy of random and incremental mutilation of the question stem appears to effectively simulate humans using different reading strategies of the questions, such as skimming or scanning. Similarly, the manipulation of the "temperature" parameter in ChatGPT LLM appears to simulate human conditions that could be influenced by emotions or other factors experienced during the taking of an exam.

These strategies allow for the simulation, using a single LLM, of a heterogeneous population responding to an exam and obtaining differential results. This population of simulated humans produced the necessary input to obtain competitive item difficulty predictions without using features extracted from the item content. These results support the idea that item difficulty lies probably more in the population answering these questions than in the content or linguistic or cognitive factors extracted from the content of the items.

References

Samah AlKhuzaey, Floriana Grasso, Terry R Payne, and Valentina Tamma. 2023. Text-based question difficulty prediction: A systematic review of automatic approaches. *International Journal of Artificial Intelligence in Education*, pages 1–53.

Luca Benedetto. 2023. A quantitative study of nlp approaches to question difficulty estimation. In *International Conference on Artificial Intelligence in Education*, pages 428–434. Springer.

Luca Benedetto, Paolo Cremonesi, Andrew Caines, Paula Buttery, Andrea Cappelli, Andrea Giussani, and Roberto Turrin. 2023. A survey on recent approaches to question difficulty estimation from text. *ACM Computing Surveys*, 55(9):1–37.

Inn-Chull Choi and Youngsun Moon. 2020. Predicting the difficulty of efl tests based on corpus linguistic features and expert judgment. *Language Assessment Quarterly*, 17(1):18–42.

George Dueñas, Sergio Jimenez, and Julia Baquero. 2015. Automatic prediction of item difficulty for short-answer questions. In *2015 10th Computing Colombian Conference (10CCC)*, pages 478–485. IEEE.

Le An Ha, Victoria Yaneva, Peter Baldwin, and Janet Mee. 2019. Predicting the difficulty of multiple choice questions in a high-stakes medical exam. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 11–20, Florence, Italy. Association for Computational Linguistics.

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.

I Lorge and L K Diamond. 1954. The prediction of absolute item difficulty by ranking and estimating techniques. *Educational and Psychological Measurement*, 14(2):365–372.

K Wauters, P Desmet, and W Van Den Noortgate. 2012. Item difficulty estimation: An auspicious collaboration between data and judgment. *Computers & Education*, 58(4):1183–1193.

Kang Xue, Victoria Yaneva, Christopher Runyon, and Peter Baldwin. 2020. Predicting the difficulty and response time of multiple choice questions using transfer learning. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 193–197, Seattle, WA, USA → Online. Association for Computational Linguistics.

Victoria Yaneva, Le An Ha, Peter Baldwin, and Janet Mee. 2020. Predicting item survival for multiple choice questions in a high-stakes medical exam. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6812–6818, Marseille, France. European Language Resources Association.

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.

Victoria Yaneva, Kai North, Peter Baldwin, An Ha Le, Saed Rezayi, Yiyun Zhou, Sagnik Ray Choudhury, Polina Harik, and Brian Clauser. 2024. Findings from the First Shared Task on Automated Prediction of Difficulty and Response Time for Multiple Choice Questions. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Using Machine Learning to Predict Item Difficulty and Response Time in Medical Tests

Mehrdad Yousefpoori-Naeim
University of Alberta / Canada
yousefpo@ualberta.ca

Shayan Zargari
University of Alberta / Canada
zargari@ualberta.ca

Zahra Hatami
Epita Engineering School / Canada
zahra.hatami@epita.fr

Abstract

Prior knowledge of item characteristics, such as difficulty and response time, without pretesting items can substantially save time and cost in high-standard test development. Using a variety of machine learning (ML) algorithms, the present study explored several (non-)linguistic features (such as Coh-Metrix indices) along with MPNet word embeddings to predict the difficulty and response time of a sample of medical test items. In both prediction tasks, the contribution of embeddings to models already containing other features was found to be extremely limited. Moreover, a comparison of feature importance scores across the two prediction tasks revealed that cohesion-based features were the strongest predictors of difficulty, while the prediction of response time was primarily dependent on length-related features.

keywords: item difficulty, response time, machine learning, Coh-Metrix, MPNet embeddings

1 Introduction

Item difficulty and response time are among the important requirements in high-standard test development. For instance, in large-scale assessment, there is often a need to develop equivalent versions of the same test to be administered to different groups of people (DePascale and Gong, 2020). When deciding on the inclusion of items in each version, it is necessary to know the difficulty level of each item and an estimate of the time needed to answer that item. Such information is traditionally gained only through pretesting (Martinková and Hladká, 2023). Pretesting, however, is not a very efficient method, as it is expensive (Antal, 2013) and raises security concerns (Settles et al., 2020). Therefore, it would be highly advantageous to devise a method to ascertain item difficulty and response time without resorting to the pretesting of items.

With this motivation, a shared task was organized as part of the Building Educational Applications (BEA) workshop at the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) 2024. The shared task invited people to develop ML models for the prediction of item difficulty and response time of a sample of 466 items from the United States Medical Licensing Examination (USMLE). The present study was conducted in relation to this shared task. For a review of the complete set of submissions to the shared task, please see Yaneva et al. (2024).

2 Related Work

In the last decade, educational assessment has witnessed a surge of interest in predicting item difficulty. Having reviewed 38 papers on item difficulty prediction, AlKhuzaey et al. (2021) provided a summary of the most frequent prediction models used, most studied domains and item types, and features with highest prediction power. ML algorithms such as neural networks and support vector machines (SVM) are commonly employed along with a variety of natural language processing (NLP) techniques used for feature extraction from text data. Language assessment was found to be the most investigated domain, and multiple-choice items were most frequently studied. A greater contribution of AlKhuzaey et al. (2021) lies in its review of the most influential features in item difficulty prediction. While most features can be categorized as either syntactic or semantic, a few studies have used psycholinguistic features (e.g., Pandarova et al., 2019), taking into account the processing of linguistic elements in the brain. The Age of Acquisition (AOA), as one of such "cognitively-motivated" features, offers an index of lexical difficulty based on how early/late in life certain words are acquired (Ha et al., 2019, p. 15). Word concreteness is

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 551–560
June 20, 2024 ©2024 Association for Computational Linguistics

another psycholinguistic feature. Concrete words are assumed to be processed faster in the brain and thus would expectedly be easier than abstract words (Brysbaert et al., 2014). The use of psycholinguistic features is not a novel approach, however. AOA and word concreteness, among similar features such as word imaginability, have long been on the list of the indices calculated by Coh-Metrix (Bruss et al., 2004). A more recent trend is the use of semantic similarity as a feature, which is discussed further in the following.

Most recently, Štěpánek et al. (2023) compared the performance of several ML algorithms in predicting the item difficulty of reading comprehension tests using features extracted from item texts. Their extracted features include word counts, word frequencies, readability indices, and lexical similarity. For lexical similarity, using Euclidean distance and cosine similarity, they calculated the textual similarity between the question and the correct option as well as between the correct option and the distractors. It was assumed that a higher similarity in the former comparison can make the question easier, while a larger similarity in the latter is associated with higher difficulty (Alsubait et al., 2014). Their results indicated that regularization-based models in general, and the elastic net (RMSE = 0.666) in particular, outperformed other models.

Although we have recently seen an increasing number of attempts to predict item characteristics such as difficulty, the wide range of test domains and other differing contextual factors make it rather difficult to make generalizations across contexts. Therefore, more studies are still needed before more valid conclusions can be drawn regarding the predictability of item characteristics. The purpose of the present study was to contribute to the line of research on predicting item characteristics in medical tests (see, for example, Xue et al., 2020, and Yaneva et al., 2021) by exploring how an assortment of linguistic and non-linguistics features can be utilized along with word embeddings to predict the item difficulty and response time of multiple-choice medical test items.

3 Methods

3.1 Corpus

The corpus of the study is a retired sample of 667 multiple-choice questions from the USMLE. The USMLE is developed by the National Board of Medical Examiners (NBME) and the Federation of State Medical Boards (FSMB) and is administered to both US and Canadian medical students. It consists of three steps, which altogether take nine hours to write. The items are written by experienced medical experts following a set of standardization guidelines. The guidelines help produce high-quality items, the difficulty of which is dependent on the difficulty of the medical content rather than any other extraneous factors.

3.2 Features

A variety of features were extracted mostly from the item stems to be used in our prediction models.

1. **Item Type:** The items in our sample of medical tests can be divided into two groups: text-only and text-and-picture items. Of the 466 items used in the train set, 10.7% (50 items) had a picture supporting the stem text. The use of pictures might help with better and faster understanding of the question.

2. **Exam Part (Step):** As mentioned in the Corpus section, the USMLE has three parts or steps. On average, Step 3 and Step 1 have the highest and lowest item difficulty, respectively. The difference between the exam steps is less considerable in terms of item response time.

3. **Stem Length:** Stem length was measured by counting the number of words in each stem. Longer stems usually take more time to read and understand, and thus they can be more difficult. The stem length of the train set items ranged from 32 to 301 words.

4. **Sentence Length Average:** A very long text can be easy to read if it contains short sentences, while a fairly short text with long sentences can be cumbersome. Therefore, we measured sentence length (as the number of words in a sentence) along with stem length.

5. **Sentence Length Maximum:** Sometimes one single lengthy (or complex) sentence can considerably interfere with comprehension, so we included Sentence Length Maximum as a separate feature in addition to Sentence Length Average.

6. **Option Count:** The higher number of answer options means a higher number of distractors, which is expected to make an item more difficult and time-consuming. Compared to the

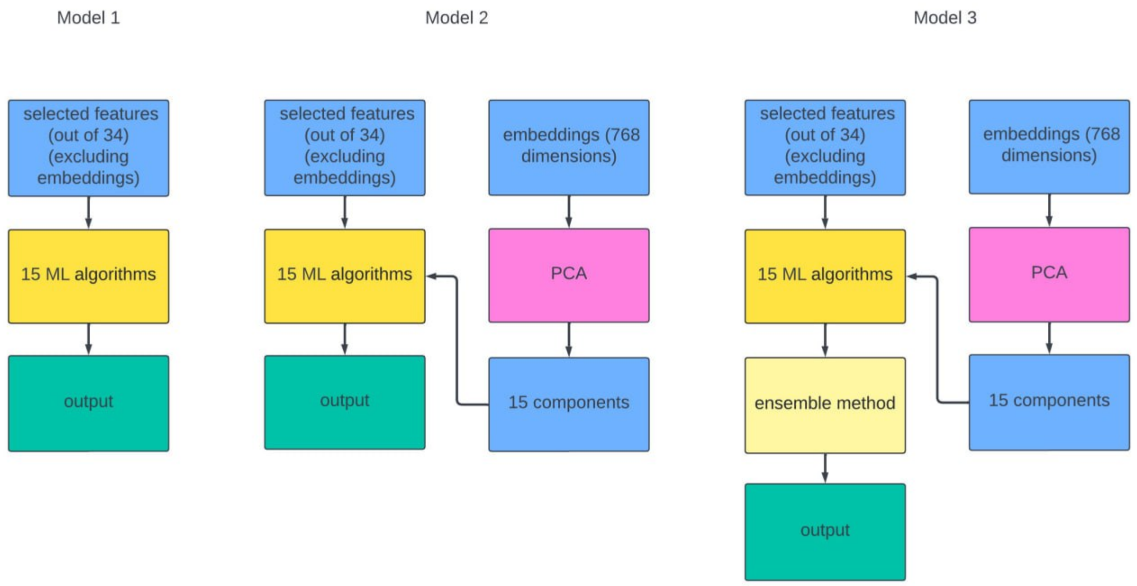

Model 1
Model 2
Model 3
selected features (out of 34) (excluding embeddings)
15 ML algorithms
output
selected features (out of 34) (excluding embeddings)
embeddings (768 dimensions)
15 ML algorithms
PCA
output
15 components
selected features (out of 34) (excluding embeddings)
embeddings (768 dimensions)
15 ML algorithms
PCA
ensemble method
15 components
output

Figure 1: Models 1-3 used for predicting difficulty and response time

minimum of four options, some items in the train set have as many as 10 options. The most common number of options is five.

7. **Challenging Topics:** Based on our observations of several highly difficult items, we formed a short list of potentially more challenging topics in our sample. The list includes the following keywords: ‘kidney’, ‘bleeding’, ‘abdominal’, ‘emergency’, ‘fever’, ‘lung’, ‘abnormality’, and ‘history’. We counted these keywords in lemmatized stems and then assigned each stem a count number accordingly. Items with higher count numbers were expected to be more difficult.

8. **Rare Words Sum:** Less frequent words are usually more difficult (Brysbaert et al., 2011). To calculate the rareness (or difficulty) of the vocabulary of item stems, we looked up each word in the BNC/COCA list (version 2.0.0), a frequency-based list of 25k English words (Nation, 2016). The BNC/COCA list classifies 25k words into 25 frequency groups based on their appearance in the two well-known corpora of BNC (British National Corpus) and COCA (Corpus of Contemporary American English).

9. **Medical Terms Sum:** We used a publicly available list of medical terms (under GNU General Public License v3.0), consisting of terms from two well-known medical dictionaries, namely OpenMedSpel by e-MedTools and Raj&Co-Med-Spel-Chek by Rajasekharan N. of Raj&Co. We counted the number of medical terms in each stem and used that as an indicator of difficulty, assuming that stems with a higher number of medical terms are more difficult and time-consuming to process. Nevertheless, it should be noted that terms can be a double-edged sword, as they can both facilitate the accessibility of information (Baleghizadeh and Yousefpoori-Naeim, 2013) and create obstacles in comprehension (Yousefpoori-Naeim et al., 2018). Moreover, not all terms in a specific domain are equal; they can be placed in a wide range of difficulty (Yousefpoori-Naeim and Baleghizadeh, 2018).

10. **Coh-Metrix Features:** Coh-Metrix is a computational tool that provides 108 indices for text analysis. These indices represent text in terms of its coherence (McNamara et al., 2014). The Coh-Metrix indices used in this study include CNCCaus, CNCTemp, CRFANPa, CRFAO1, CRFCWO1, DESWLlt, LDTTRc, LDVOCDa, LSAGN, LSASSpd, PCCNCz, PCCONNz, PCDCz, PCREFz, PCSYNz, PCTEMPz, RDFRE, SMCAUSlsa, WRDADJ, WRDADV, WRDFRQa, WRDMEAc, WRDNOUN, and WRDVERB. The complete names of these features are provided in Table 2 in the appendix. For more informa-

tion on what each of these features refers to and how they are calculated, see Coh-Metrix version 3.0 indices.

11. **Embeddings:** We used the MPNet encoder to obtain embeddings for each stem text. MPNet is a pre-trained transformer-based language model, which has been shown to outperform similar well-known pre-trained models, such as BERT and RoBERTa, in several tasks (Song et al., 2020). MPNet encoder generates embeddings in the form of 768-dimension vectors. The embeddings represent text in various aspects, including its context, meaning, and syntactic structure.

3.3 Algorithms

We deployed 15 ML algorithms to achieve the highest performance: Linear Regression, Ridge Regression, Lasso Regression, ElasticNet, Stochastic Gradient Descent (SGD), Support Vector Regression (SVR), Decision Tree, Random Forest, Gradient Boosting, Extra Trees, AdaBoost, K-Neighbors, Multilayer Perceptron (MLP), XGBoost, and CatBoost. These algorithms cover a broad spectrum of ML techniques, each with its own strengths and use cases.

3.4 Procedures

Irrespective of the algorithm used, three models were built for each prediction task incrementally. First, a selection of features excluding embeddings was used to train Model 1. Next, embeddings were added to build Model 2. Finally, an ensemble method was utilized to find the best combination of algorithms to be used in Model 3. Figure 1 depicts the structure of the three models in more detail.

Given the high number of our features, we made attempts at different stages of the models to filter out the less relevant features, as feature reduction can enhance model efficiency and lower the risk of overfitting (Ying, 2019). Initially, using a heat map, we detected instances of high correlation in every possible pair of features to address multicollinearity. We marked a correlation coefficient of 0.8 and higher as the presence of multicollinearity (Hae, 2019) and removed one of the two features in the pair. The choice of features for removal was based on theoretical justification and/or literature insights. In a later stage, after Model 1 was initially trained, we gradually removed a few more features based on feature importance results and retrained the model with the truncated list of features. If model performance remained relatively stable, we kept the removed features out of the feature set; otherwise, we re-inserted the removed features one by one to reach comparable performance results. The final lists of selected features used for each prediction task are provided in Table 3 in the appendix. The feature of embeddings went through a reduction process as well. Principal component analysis (PCA) was used to reduce the 768 dimensions of embeddings to 15 components. This number of components was chosen after experimenting with a range of components from 5 to 20, with 15 components yielding the best result.

Cross-validation (CV) was utilized to make the best of the limited data. After randomly leaving 20% of the data out for testing the final models, we ran a 5-fold cross-validation on the remaining 80% subset. Root mean square error (RMSE) results were reported on both the test set and the five folds of the CV subset. A comparison of model performance in training and test sets helps with the detection of overfitting (Ying, 2019).

RMSE was used as the main evaluation metric in the study. It is calculated based on the following formula:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}} \tag{1}$$

where y_i is the actual outcome value and $\hat{y}_i$ is the predicted one for the i-th data, with N denoting the total number of data. RMSE is thus an indicator of the prediction error, i.e., the difference between predicted and actual outcome values. Lower RMSE values indicate lower prediction error.

4 Results

Table 1 presents the RMSE results of all three models in the test and CV subsets for both prediction tasks. In both tasks, Model 2 has a marginally better performance (i.e., lower RMSE) than Model 1, indicating that the addition of embeddings only slightly enhances model performance. Additionally, using the ensemble method (Model 3) did not lead to any performance improvement in either of the tasks.

Unlike the RMSE results, the feature importance results were relatively different in the two prediction tasks. In particular, Coh-Metrix features had a stronger presence in the top features for the difficulty task. In predicting difficulty, PCTEMPz,

Table 1: Model comparisons for predicting difficulty and response time

Models	Difficulty			Response Time		
	Method	Test RMSE	CV RMSE	Method	Test RMSE	CV RMSE
Model 1	CatBoost	0.277	0.314	K-Neighbors	23.743	25.586
Model 2	AdaBoost	0.269	0.315	K-Neighbors	23.271	24.898
Model 3	Ensemble*	0.269	0.315	Ensemble**	23.271	24.898

*{AdaBoost} **{K-Neighbors}

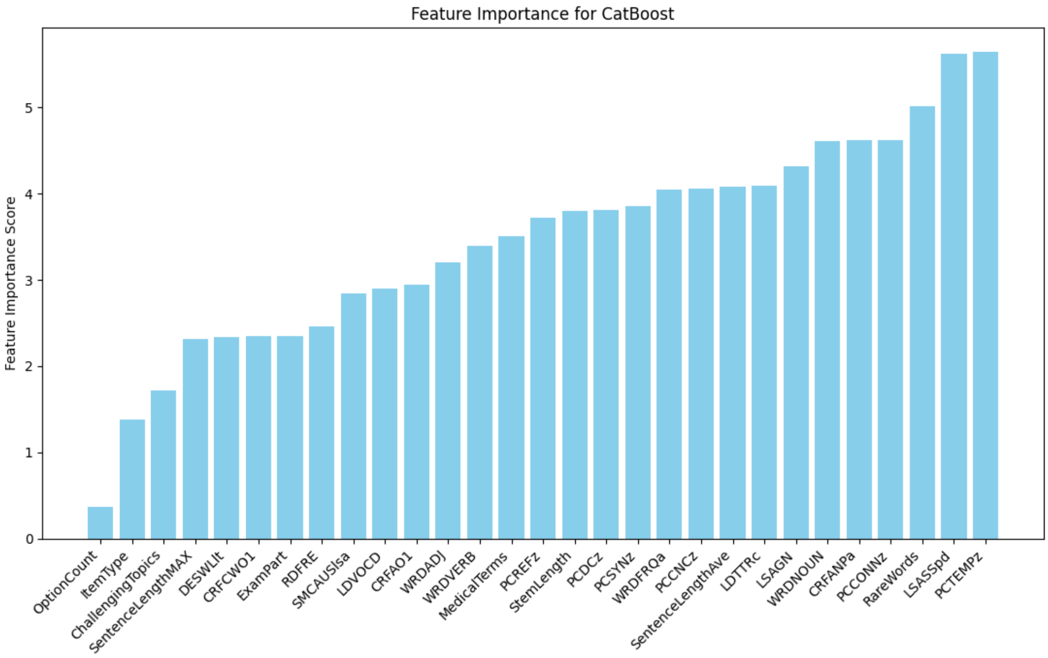


Figure 2: Feature importance scores for predicting difficulty using the CatBoost method

LSASSpd, and Rare Words Sum are the top three features (Figure 2), while Sentence Length Max, Stem Length, and Medical Terms Sum stand out as the top three features predicting response time (Figure 3). Moreover, unlike the prediction task of difficulty, a few Coh-Metrix features were found to have a weak negative relationship with response time.

5 Discussion

A comparison of the RMSE results across the three models in both prediction tasks indicates that the addition of embeddings (i.e., Model 2) had a very small contribution to model performance. While this finding was against our initial expectation, it does bear credence when taking into account the large number of features already fed into Model 1. The selected Coh-Metrix indices coupled with our extracted features (such as Rare Words Sum and Medical Terms Sum) captured most of the variance, leaving not much else to be explained by embeddings. A similar scenario has been present in some other studies. In (Ha et al., 2019), for example, adding either Word2vec or ELMo embeddings to a list of various linguistic features improved RMSE results by minimal margins.

As for Model 3 in both prediction tasks, the ensemble method was ineffective in reducing RMSE values because there was no possible combination of algorithms that would result in a better model performance. In both tasks, the difference between the top-performing algorithm and the rest of the algorithms was wide; therefore, combining the top algorithm with any other one would only harm the performance. Another reason could be that the algorithms are making similar predictions, meaning

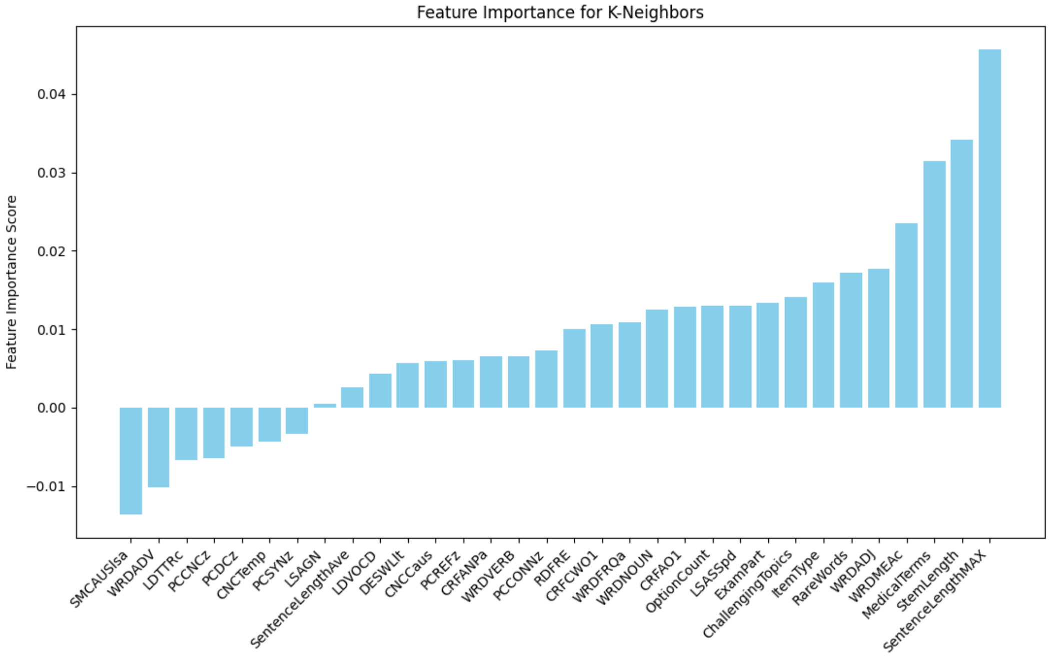


Figure 3: Feature importance scores for predicting response time using the K-Neighbors method

that there is a high correlation between their predictions. The ensemble method usually works best when models trained by different algorithms have different strengths and weaknesses, so combining models could lead to one model compensating for deficiencies in another.

The feature importance scores exhibited dissimilar patterns in the two prediction tasks. Features measuring the cohesion of the stem text played a major role in predicting difficulty: The vast majority of the top predictors of difficulty are cohesion-based Coh-Metrix features. On the other hand, non-Coh-Metrix features, especially length-related ones, constituted the main group of predictors of item response. Length, measured as either the maximum number of words in a sentence (i.e., Sentence Length Max) or the total number of words in the stem text (i.e., Stem Length), is the predominant predictor of item response. Compared to difficulty, response time can be considered less complicated to explain, as it is highly dependent on simple length-related features.

6 Limitations

Two limitations need to be taken into account when interpreting the results of the study. Firstly, the quality of extracted features was dependent on the quality of stem text preprocessing. While preprocessing text data (e.g., tokenization and lemmatization) is generally challenging, the text of medical items can pose additional challenges. The stem of many medical items typically contains a tabulation of data, e.g., laboratory results and blood pressure measures. When embedded within the text, such data can negatively impact the accuracy of feature calculations. For example, a list of items and numbers within a syntactically simple sentence can make it appear as a complex sentence in measures of sentence complexity. It can also interfere with coherence measures calculated through the Coh-Matrix.

The second limitation concerns the results of feature importance. Different algorithms may produce relatively different feature importance sets as they try to reach their highest prediction accuracy. Therefore, the top three or five features in one algorithm can differ from those in another algorithm even with a very close RMSE. To better understand the contribution of each feature to the prediction model, experimental studies are recommended, as the direct effect of individual variables can be more reliably examined through experimental control and manipulation (Yousefpoori-Naeim et al., 2023).

7 Conclusion

The present study explored a selection of diverse features to predict the difficulty and response time of a sample of multiple-choice medical test items using a variety of ML algorithms. In either of the prediction tasks, the addition of embeddings to the list of features did not make a considerable contribution to model performance, and the use of the ensemble method was not effective either. In feature importance scores, however, the two tasks showed dissimilar patterns. Features measuring cohesion were especially effective in predicting difficulty, while length-related features were the main predictors of response time.

While future studies can examine the role of many other features in predicting item characteristics of medical tests, we would like to draw attention to collecting data from item writers to be used as a potential feature. Especially in the case of item difficulty, medical test writers can be asked to rate the difficulty of the items they develop. While students might perceive items differently from what test writers would assume, item writers' ratings could still correlate highly with actual difficulty values. This feature enjoys high practicality and low cost, as item writers can give difficulty ratings as they write their own items. A more advanced, but also more expensive approach is to have item writers rate each others' items as well.

References

Samah AlKhuzaey, Floriana Grasso, Terry R. Payne, and Valentina Tamma. 2021. A systematic review of data-driven approaches to item difficulty prediction. In *Artificial Intelligence in Education*, pages 29–41, Cham. Springer International Publishing.

Tahani Alsubait, Bijan Parsia, and Uli Sattler. 2014. Generating multiple choice questions from ontologies: Lessons learnt. In *OWLED*, pages 73–84. Citeseer.

Margit Antal. 2013. On the use of elo rating for adaptive assessment. *Studia Universitatis Babes-Bolyai, Informatica*, 58(1):29–41.

Sasan Baleghizadeh and Mehrdad Yousefpoori-Naeim. 2013. Surveying metalanguage through three efl textbooks. *E-International Journal of Educational Research*, 4(3):27–40.

Michell Bruss, Michael J. Albers, and Danielle McNamera. 2004. Changes in scientific articles over two hundred years: a coh-metrix analysis. In *Proceedings of the 22nd Annual International Conference on Design of Communication: The Engineering of Quality Documentation*, SIGDOC '04, page 104–109, New York, NY, USA. Association for Computing Machinery.

Marc Brysbaert, Matthias Buchmeier, Markus Conrad, Arthur M. Jacobs, Jens Bölte, and Andrea Böhl. 2011. The word frequency effect. *Experimental Psychology*, 58(5):412–424. PMID: 21768069.

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior research methods*, 46:904–911.

Charles DePascale and Brian Gong. 2020. Comparability of individual students' scores on the "same test. *Comparability of large-scale educational assessments: Issues and recommendations*, pages 25–48.

Le An Ha, Victoria Yaneva, Peter Baldwin, and Janet Mee. 2019. Predicting the difficulty of multiple choice questions in a high-stakes medical exam. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 11–20, Florence, Italy. Association for Computational Linguistics.

Kim Jong Hae. 2019. Multicollinearity and misleading statistical results. *kja*, 72(6):558–569.

Patrícia Martinková and Adéla Hladká. 2023. *Computational aspects of psychometric methods: With R*. CRC Press.

Danielle S McNamara, Arthur C Graesser, Philip M McCarthy, and Zhiqiang Cai. 2014. *Automated evaluation of text and discourse with Coh-Metrix*. Cambridge University Press.

Ian Stephen Paul Nation. 2016. *Making and using word lists for language learning and testing*. John Benjamins Publishing Company.

Irina Pandarova, Torben Schmidt, Johannes Hartig, Ahcène Boubekki, Roger Dale 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*, 29:342–367.

Burr Settles, Geoffrey T. LaFlair, and Masato Hagiwara. 2020. Machine Learning–Driven Language Assessment. *Transactions of the Association for Computational Linguistics*, 8:247–263.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *CoRR*, abs/2004.09297.

Kang Xue, Victoria Yaneva, Christopher Runyon, and Peter Baldwin. 2020. Predicting the difficulty and response time of multiple choice questions using transfer learning. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 193–197, Seattle, WA, USA → Online. Association for Computational Linguistics.

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.

Victoria Yaneva, Kai North, Peter Baldwin, An Ha Le, Saed Rezayi, Yiyun Zhou, Sagnik Ray Choudhury, Polina Harik, and Brian Clauser. 2024. Findings from the First Shared Task on Automated Prediction of Difficulty and Response Time for Multiple Choice Questions. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Xue Ying. 2019. An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, 1168(2):022022.

Mehrdad Yousefpoori-Naeim and Sasan Baleghizadeh. 2018. Towards finding a difficulty index for english grammatical terminology. *Terminology*, 24(2):236–261.

Mehrdad Yousefpoori-Naeim, Okan Bulut, and Bin Tan. 2023. Predicting reading comprehension performance based on student characteristics and item properties. *Studies in Educational Evaluation*, 79:101309.

Mehrdad Yousefpoori-Naeim, Lawrence Jun Zhang, and Sasan Baleghizadeh. 2018. Resolving the terminological mishmash in teaching link words in efl writing. *Chinese Journal of Applied Linguistics*, 41(3):321–337.

Lubomír Štěpánek, Jana Dlouhá, and Patrícia Martinková. 2023. Item difficulty prediction using item text features: Comparison of predictive performance across machine-learning algorithms. *Mathematics*, 11(19).

A Appendix

Table 2: Coh-Metrix feature labels and their descriptions

Feature Label	Description
CNCCaus	Causal connectives incidence
CNCTemp	Temporal connectives incidence
CRFANPa	Anaphor overlap, all sentences
CRFAO1	Argument overlap, adjacent sentences binary, mean
CRFCWO1	Content word overlap, adjacent sentences proportional, mean
DESWLlt	Word length, number of letters, mean
LDTTRc	Lexical diversity, type-token ratio, content word lemmas
LDVOCDa	Lexical diversity, VOCD, all words
LSAGN	LSA given/new, sentences, mean
LSASSpd	LSA overlap, all sentences in a paragraph, standard deviation
PCCNCz	Text Easability, PC Word concreteness, z score
PCCONNz	Text Easability, PC Connectivity, z score
PCDCz	Text Easability, PC Deep cohesion, z score
PCREFz	Text Easability, PC Referential cohesion, z score
PCSYNz	Text Easability, PC Syntactic simplicity, z score
PCTEMPz	Text Easability, PC Temporality, z score
RDFRE	LSA verb overlap
SMCAUSlsa	Flesch Reading Ease
WRDADJ	Adjective incidence
WRDADV	Adverb incidence
WRDFRQa	CELEX Log frequency for all words, mean
WRDMEAc	Meaningfulness, Colorado norms, content words, mean
WRDNOUN	Noun incidence
WRDVERB	Verb incidence

Table 3: List of features used in each prediction task

Features	**Difficulty**	**Item Response**
Item Type	●	●
Exam Part (Stem)	●	●
Stem Length	●	●
Sentence Length Average	●	●
Sentence Length Maximum	●	●
Option Count	●	●
Challenging Topics	●	●
Rare Words Sum	●	●
Medical Terms Sum	●	●
CNCCaus		●
CNCTemp		●
CRFANPa	●	●
CRFAO1	●	●
CRFCWO1	●	●
DESWLlt	●	●
LDTTRc	●	●
LDVOCDa	●	●
LSAGN	●	●
LSASSpd	●	●
PCCNCz	●	●
PCCONNz	●	●
PCDCz	●	●
PCREFz	●	●
PCSYNz	●	●
PCTEMPz	●	
RDFRE	●	●
SMCAUSlsa	●	●
WRDADJ	●	●
WRDADV		●
WRDFRQa	●	●
WRDMEAc		●
WRDNOUN	●	●
WRDVERB	●	●

Large Language Model-based Pipeline for Item Difficulty and Response Time Estimation for Educational Assessments

Hariram Veeramani [1]
hariram@ucla.edu

Surendrabikram Thapa [2]
surendrabikram@vt.edu

Natarajan Balaji Shankar [1]
balaji1312@ucla.edu

Abeer Alwan [1]
alwan@ee.ucla.edu

[1] **University of California, Los Angeles**

[2] **Virginia Tech**

Abstract

This work presents a novel framework for the automated prediction of item difficulty and response time within educational assessments. Utilizing data from the BEA 2024 Shared Task, we integrate Named Entity Recognition, Semantic Role Labeling, and linguistic features to prompt a Large Language Model (LLM). Our best approach achieves an RMSE of 0.308 for item difficulty and 27.474 for response time prediction, improving on the provided baseline. The framework's adaptability is demonstrated on audio recordings of 3rd-8th graders from the Atlanta, Georgia area responding to the Test of Narrative Language. These results highlight the framework's potential to enhance test development efficiency.

1 Introduction

Standardized tests are essential tools for evaluating knowledge and ability for academic and professional purposes, and thus must be rigorously designed and meet stringent criteria. Key aspects include diverse item difficulty for comprehensive skill evaluation and appropriate response time allocation – insufficient time compromises fairness, while excessive time leads to inefficiencies (Huggins-Manley et al., 2022). Traditionally, item difficulty and response time optimization have relied on *pretesting*, where new items are embedded in live exams. However, this labor-intensive process limits the number of new items and introduces security risks through potential overuse (Settles et al., 2020). In high-stakes examinations like the United States Medical Licensing Examination (USMLE) [1], these challenges necessitate the exploration of alternative approaches for more secure and efficient test design.

In response to these challenges, recent research explores automated prediction using the text of items themselves. This approach promises to streamline test development, enhance exam fairness, and mitigate security risks associated with item overexposure. The automated prediction of *item difficulty* and *item response time* shared task at the 19th BEA Workshop aims to address this gap (Yaneva et al., 2024). Advancements in Large Language Models (LLMs), trained on massive text corpora, hold significant potential for discerning language patterns indicative of item difficulty and response time. This paper outlines our methodology for automated prediction of these characteristics, leveraging named entity recognition, semantic role labeling, and LLMs. We further evaluate the framework's validity across modalities by applying it to a dataset of children's oral responses to the Test of Narrative Language. Our approach integrates these technologies to analyze the complexities of test item texts, aiming to accurately predict both difficulty level and response time.

2 Related Works

In recent years, the prediction of item difficulty and response time has garnered significant attention in the field of educational assessment research. Prior work in this field employed techniques rooted in classical test theory and item response theory. More recently, the advent of sophisticated machine learning approaches has enabled novel methods for modeling these parameters (Yaneva et al., 2020, 2021).

In Lin et al. (2019) an LSTM-based method for Chinese reading comprehension tests was proposed. It achieved high accuracy utilizing word embeddings and text correlation networks. Similarly, Hochreiter and Schmidhuber (1997) employed word embeddings within a semantic space to analyze relationships between multiple-choice test components, finding correlations between semantic similarity and item difficulty. Research on item difficulty prediction in medical exams has also advanced significantly with Qiu et al. (2019) introducing the Document enhanced Attention based

[1] https://www.usmle.org/

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 561–566
June 20, 2024 ©2024 Association for Computational Linguistics

neural Network (DAN) framework using semantic relevance and similarity for difficulty assessment. Ha et al. (2019) further demonstrated that embeddings and linguistic features extracted from test documents outperform simple text complexity measures in predicting construct-relevant difficulty in MCQs. Baldwin et al. (2021) incorporated item response time prediction, emphasizing the importance of understanding how test-takers interact with items. In a similar vein, Xue et al. (2020) found transfer learning beneficial for USMLE item difficulty prediction, suggesting stems alone are optimal for difficulty, while the entire question benefits response time prediction. Despite these advancements, the joint prediction of item difficulty and response time remains under-explored, motivating our proposed technique designed to address this gap.

3 Data

We evaluate our framework primarily on the 2024 BEA shared task dataset constructed from the USMLE. As an auxiliary task, we also test its validity on the Test of Narrative Language.

3.1 Shared Task Description

The BEA 2024 Shared Task focuses on the automated prediction of item difficulty and item response time for standardized exams, with an emphasis on the USMLE. This task seeks to enhance the fairness and validity of standardized exams by streamlining the estimation of item characteristics, reducing the reliance on extensive pretesting. The shared task comprises two tracks:

- **Track 1: Item Difficulty Prediction** predicts the difficulty level of test items using item text and relevant metadata.
- **Track 2: Item Response Time Prediction** predicts the average time required by test-takers to answer an item utilizing item text and metadata.

3.1.1 Dataset

This task utilizes a dataset of 667 retired questions from USMLE Steps 1, 2 CK (Clinical Knowledge), and 3. These items cover a range of medical knowledge and were authored by experts. The dataset includes the following components for each item:

- **Item Text (Stem)**: Clinical scenario/question presented.
- **Answer Options**: Response choices (A-J, some items may have fewer options).
- **Correct Answer (Key)**: Correct response letter.
- **Item Type**: Indicates text-only or image-based (images not provided).
- **Exam Step**: Which USMLE step the item belongs to.
- **Item Difficulty**: Numerical difficulty value (higher=more difficult).
- **Response Time**: Average response time (seconds) from live exam data.

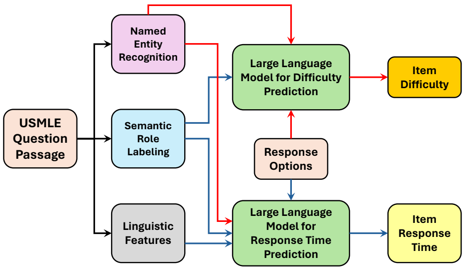


Figure 1: The proposed framework for item difficulty and response time prediction

3.2 Test of Narrative Language (TNL)

This work also uses audio recordings of 185 3rd-8th grade students from the Atlanta, Georgia, area as they perform the “Test of Narrative Language (TNL)" assessment (data collected in Fisher et al. (2019)). In "Task 2 - Picture Description" in the TNL, the children were shown an image containing a character and several elements to describe. The students were then asked to tell a story about the image, making their story as complete as possible. Each child’s response to the prompt was recorded, and each child, on average, took about 3 minutes to complete their story. Each child’s assessment was administered and audio recorded by a trained member of the project staff according to the TNL protocols. Recordings were then independently scored by two speech-language pathologists. If disagreements occurred in scoring, the two scorers reviewed the audio and discussed differences to reach a consensus. Each child’s score was an integer value between 0 and the total number of test keywords. Recordings were taken at the child’s school. Audio was recorded in stereo at a sampling rate of 48kHz. All recordings were resampled to mono audio with a sampling rate of 16kHz for experimentation.

3.3 Evaluation

The evaluation for both tracks of the shared task, and the Test of Narrative Language, is based on the Root Mean Squared Error (RMSE) metric, offering an objective measure of the accuracy of predictions made by the proposed pipeline.

4 Methodology

4.1 Item Difficulty Prediction

Our item difficulty prediction methodology integrates multiple advanced NLP techniques to enhance the precision of our predictions. We outline our approach in three main steps: Named Entity Recognition (NER), Semantic Role Labeling (SRL), and the final difficulty prediction.

4.1.1 Named Entity Recognition

For Named Entity Recognition (NER), we employ a dual-model strategy using both Longformer (Beltagy et al., 2020) and a choice between three Large Language Models (LLMs), Mistral-7B (Jiang et al., 2023), Llama-7B (Touvron et al., 2023), or Gemma-7B (Team et al., 2023) to extract named entities from the entire question text. For LLMs, we provide input as the question and specifically prompt them as follows: `Understand the input sentence and annotate the named entities from the Input Context`. This process can be represented as follows:

$$NER_{longformer} = Extract_{longformer}(Question)$$

$$NER_{LLM} = Extract_{LLM}(Question)$$

$$NER_{Union} = NER_{longformer} + NER_{LLM}$$

This process yields three combinations of NER outputs, one for each LLM, by taking the union of NERs extracted from Longformer and the selected LLM. This approach ensures a more comprehensive and accurate set of named entities by leveraging the strengths of each model.

4.1.2 Semantic Role Labeling

Following Named Entity Recognition (NER), we employ Semantic Role Labeling (SRL) utilizing both AllenNLP SRL Model (BERT Variant) (Gardner et al., 2018) and the selected LLM. SRL functions to identify semantic relationships within the sentence, attributing roles to entities according to their contextual significance. For SRL, the process is analogous to that of NER, employing both AllenNLP SRL and LLM to analyze the text. This process can be represented as:

$$SRL_{BERT} = Analyze_{BERT}(Question)$$

$$SRL_{LLM} = Analyze_{LLM}(Question)$$

$$SRL_{Union} = SRL_{BERT} + SRL_{LLM}$$

For LLMs to generate SRL, we provide the question and specifically prompt them as follows: `Understand the input context, which consists of the input sentence and the associated named entities, then annotate the semantic role labels of the input context`. This step deepens our pipeline's comprehension of the question's structure and content, thus facilitating more precise predictions of item difficulty.

4.1.3 Difficulty Prediction

Finally, we integrate NER and SRL outputs to predict item difficulty. The LLM is prompted to estimate difficulty based on the complexity of relating the correct answer to the identified entities and their semantic roles.

$$Difficulty = Predict_{LLM}(NER_{union}, SRL_{output})$$

We prompt the LLMs by providing input as the question, NER, SRL, answer, and the prompt as: `For answer option set, understand the input context consisting of an input sentence, a collection of named entities and semantic role information, summarize the association with the` i`th answer option. Depending on the difficulty level of the linkages between input context and [answer options], assign the input context a score in the range of 0 to 1.4.` This approach leverages the LLM's language understanding capabilities, enriched by the detailed insights from NER and SRL, enabling a more informed prediction of item difficulty.

4.2 Item Response Time Prediction

For item response time prediction, as shown in Fig. 1, we use linguistic features in addition to the NER and SRL features. For NER and SRL features, we follow the same steps as for the difficulty prediction subtask.

4.2.1 Linguistic Features from Question

For item response time prediction, we begin by extracting a subset of the 255 hand-crafted linguistic features from LingFeat (Lee et al., 2021). Among all features, we only take numerical and syntactic features. The LLM is then prompted to estimate response time using the question, NER, SRL, answer, linguistic features and the following prompt: `For answer option set, understand the input context consisting of an input sentence, a collection of named entities, semantic role information, Concatenate lingfeat numerical and syntactic features to summarize the association with the` i`th answer option. Depending on the exhaustiveness of the linkages`

```
demonstrated with input context and
[answer options], assign the entire input
context a response time in the range
of 25.0 to 230.00. Higher value would
indicate longer response time and higher
exhaustiveness.
```

Both item difficulty and response time predictions are performed utilizing the Langchain library (LangChain, 2024) for chaining API calls to the LLM models in different stages, as well as to post-process the outputs after each stage.

4.3 Difficulty and Item Response Time for Oral Assessments

For recordings from the Test of Narrative Language, we first generate Automatic Speech Recognition (ASR) transcripts using the Whisper model (Radford et al., 2023) as in Veeramani et al. (2023). Prior studies on literacy development (MEIERS and MENDELOVITS, 2016), highlight the role played by item response theory in measuring narrative proficiency and literacy among school children. Item difficulty is assessed utilizing two metrics: 1) Transcription Word Accuracy: Calculated as described in Oliveira et al. (2022). 2) Proportion of Correct Responses: We measure the percentage of children who correctly answer a test item, providing an additional indicator of item difficulty. To model item response time, we analyze the time taken by disfluencies exhibited by speakers during the assessment. These disfluencies, classified as filled pauses (FP), partial words (PW), repetitions (RP), revisions (RV), and restarts (RS), are extracted using models pretrained on the Switchboard corpus (Godfrey et al., 1992) following the methodology outlined in Romana et al. (2023).

4.4 System Design

As per the BEA 2024 Shared Task guidelines, we attempt the item difficulty and response time prediction task with three separate pipelines. The runs use identical pipelines and differ only in the choice of the LLM, with Run 1 using Llama2-7B, Run 2 Mistral-7B, and Run 3 using Gemma-7B.

5 Results and Discussion

5.1 BEA 2024 Shared Task

We first report our results on the BEA 2024 shared task, comparing the baseline with three variants of our proposed pipeline. Our findings (Table 1) demonstrate that prompting Llama2-7B (Run 1) for simultaneous prediction of response time and item difficulty outperforms the DummyRegressor baseline and other LLMs. Similarly, Gemma-7B (Run 3) also exceeds the baseline. We did not perform any ablation studies. However, these results align with prior research on LLM reasoning capabilities (Johnson et al., 2023), supporting the value of our chosen handcrafted features as supplementary input.

Table 1: RMSE values of different runs on the BEA 2024 Shared Task. Numbers in bold represent best results

Method	Item Difficulty	Response Time
Baseline	0.311	31.68
Run 1	**0.308**	**27.474**
Run 2	0.329	31.962
Run 3	**0.308**	28.191

5.2 Test of Narrative Language

Table 2: RMSE values from different runs on the TNL - Task 2 data

Method	Item Difficulty	Response Time
Baseline	4.043	4.941
Run 1	2.162	2.038
Run 2	2.0578	2.0237
Run 3	**2.007**	**2.022**

As shown in Table 2, Gemma-7B (Run 3) demonstrates superior performance in predicting both response time and item difficulty, exceeding the baseline and other LLMs. Similar to the results seen in the BEA 2024 Shared Task, the inclusion of numerical, lexical, and linguistic features likely aides in understanding the complex interplay of within the input and the syntactic/semantic relationships needed to correctly identify the answer.

5.3 Conclusion

This paper introduces a novel framework for automating the prediction of item difficulty and response time, a crucial aspect of educational assessment design. Our system, utilizing Named Entity Recognition, Semantic Role Labeling, and linguistic features in conjunction with a Large Language Model, demonstrates promising performance on the BEA 2024 Shared Task data, achieving RMSE values of 0.308 (item difficulty), and 27.474 (item response time). The framework's adaptability was further evidenced by its successful application to audio recordings from the Test of Narrative Language, highlighting the potential of this approach to streamline test development.

Limitations

While promising, our framework has limitations: **Model Interpretability:** The LLM's decision-making process lacks transparency. Future research should explore methods for increasing interpretability and providing human-understandable explanations.

Linguistic Feature Scope: Our current implementation analyzes a specific set of linguistic features for item response time prediction. It is possible that additional features, such as specific domain-related vocabulary, could further enhance prediction accuracy.

Domain Specificity: While our framework shows promise for both written and oral assessments, its performance may vary across different domains and test formats. Further research is needed to evaluate and potentially adapt the framework for optimal performance in specific testing contexts.

Addressing these limitations will improve the framework's accuracy, efficiency, and fairness in educational assessments.

Ethics Statement

We offer a brief discussion of the licensing requirements for the models and datasets used in our submission.

Datasets: The USMLE dataset employed for item difficulty and response time prediction is provided by the BEA Shared Task. The auxiliary data from the Test of Narrative Language is derived from a copyrighted assessment. Data from individual participants is not publicly released to protect test-taker anonymity.

Pretrained Models: The Longformer, BERT, Mistral, and Whisper models utilized in this work are released under an Apache-2.0 license. Gemma and Llama2 are available for use with a custom license permitting non-commercial use.

Acknowledgements

We would like to thank Prof. Robin Morris for collecting and labeling the audio recording from the Test of Narrative Language used as part of this study. This work was also funded in part by the National Science Foundation.

References

Peter Baldwin, Victoria Yaneva, Janet Mee, Brian E Clauser, and Le An Ha. 2021. Using natural language processing to predict item response times and improve test construction. *Journal of Educational Measurement*, 58(1):4–30.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Evelyn L Fisher et al. 2019. Executive Functioning and Narrative Language in Children with Dyslexia. *American Journal of Speech-Language Pathology*, 28(3):1127–1138.

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.

John J Godfrey, Edward C Holliman, and Jane McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *Acoustics, speech, and signal processing, ieee international conference on*, volume 1, pages 517–520. IEEE Computer Society.

Le An Ha, Victoria Yaneva, Peter Baldwin, and Janet Mee. 2019. Predicting the difficulty of multiple choice questions in a high-stakes medical exam. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 11–20, Florence, Italy. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

A Corinne Huggins-Manley, Brandon M Booth, and Sidney K D'Mello. 2022. Toward argument-based fairness with an application to ai-enhanced educational assessments. *Journal of Educational Measurement*, 59(3):362–388.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Alexander Johnson, Hariram Veeramani, Natarajan Balaji Shankar, and Abeer Alwan. 2023. An Equitable Framework for Automatically Assessing Children's Oral Narrative Language Abilities. In *Proc. INTERSPEECH 2023*, pages 4608–4612.

LangChain. 2024. [link].

Bruce W Lee, Yoo Sung Jang, and Jason Hyung-Jong Lee. 2021. Pushing on text readability assessment: A transformer meets handcrafted linguistic features. *arXiv preprint arXiv:2109.12258*.

Li-Huai Lin, Tao-Hsing Chang, and Fu-Yuan Hsu. 2019. Automated prediction of item difficulty in reading comprehension using long short-term memory. In *2019 international conference on asian language processing (ialp)*, pages 132–135. IEEE.

MARION MEIERS and JULIETTE MENDELOVITS. 2016. A longitudinal study of literacy development in the early years of school. *UNDERSTANDING WHAT WORKS IN ORAL READING ASSESSMENTS*, page 118.

Chaina S Oliveira, João VC Moraes, Telmo Silva Filho, and Ricardo BC Prudêncio. 2022. A two-level item response theory model to evaluate speech synthesis and recognition. *Speech Communication*, 137:19–34.

Zhaopeng Qiu, Xian Wu, and Wei Fan. 2019. Question difficulty prediction for multiple choice problems in medical exams. In *Proceedings of the 28th acm international conference on information and knowledge management*, pages 139–148.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning, ICML 2023*, volume 202, pages 28492–28518. PMLR.

Amrit Romana, Kazuhito Koishida, and Emily Mower Provost. 2023. Automatic disfluency detection from untranscribed speech. *arXiv preprint arXiv:2311.00867*.

Burr Settles, Geoffrey T. LaFlair, and Masato Hagiwara. 2020. Machine learning–driven language assessment. *Transactions of the Association for Computational Linguistics*, 8:247–263.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

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*.

Hariram Veeramani, Natarajan Balaji Shankar, Alexander Johnson, and Abeer Alwan. 2023. Towards Automatically Assessing Children's Oral Picture Description Tasks. In *Proc. 9th Workshop on Speech and Language Technology in Education (SLaTE)*, pages 119–120.

Kang Xue, Victoria Yaneva, Christopher Runyon, and Peter Baldwin. 2020. Predicting the difficulty and response time of multiple choice questions using transfer learning. In *Proceedings of the fifteenth workshop on innovative use of NLP for building educational applications*, pages 193–197.

Victoria Yaneva, Le An Ha, Peter Baldwin, and Janet Mee. 2020. Predicting item survival for multiple choice questions in a high-stakes medical exam. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6812–6818, Marseille, France. European Language Resources Association.

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.

Victoria Yaneva, Kai North, Peter Baldwin, An Ha Le, Saed Rezayi, Yiyun Zhou, Sagnik Ray Choudhury, Polina Harik, and Brian Clauser. 2024. Findings from the First Shared Task on Automated Prediction of Difficulty and Response Time for Multiple Choice Questions. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

UNED team at BEA 2024 Shared Task: Testing different Input Formats for predicting Item Difficulty and Response Time in Medical Exams

Alvaro Rodrigo, Sergio Moreno-Álvarez, Anselmo Peñas
NLP & IR group at UNED
Madrid, Spain
{alvarory,smoreno,anselmo}@lsi.uned.es

Abstract

This paper presents the description and primary outcomes of our team's participation in the BEA 2024 shared task. Our primary exploration involved employing transformer-based systems, particularly BERT models, due to their suitability for Natural Language Processing tasks and efficiency with computational resources. We experimented with various input formats, including concatenating all text elements and incorporating only the clinical case. Surprisingly, our results revealed different impacts on predicting difficulty versus response time, with the former favoring clinical text only and the latter benefiting from including the correct answer. Despite moderate performance in difficulty prediction, our models excelled in response time prediction, ranking highest among all participants. This study lays the groundwork for future investigations into more complex approaches and configurations, aiming to advance the automatic prediction of exam difficulty and response time.

1 Introduction

In this paper, we describe the proposals sent by our team to the BEA 2024 shared task (Yaneva et al., 2024). This task aims to predict standardized exams' difficulty (Track 1) and response time (Track 2). The data used in this task is from a high-stakes medical exam called the United States Medical Licensing Examination[1]. The exams are provided in a multiple-choice format, with answer candidates ranging from 4 to 10.

Adjusting the difficulty of exams to align with the intended level of evaluation is crucial for ensuring the validity and fairness of assessments. Educators can accurately gauge students' understanding and proficiency within the targeted subject matter by calibrating the difficulty appropriately. This practice also promotes an equitable assessment environment where students can handle their challenges, allowing for a more reliable measure of their knowledge and skills. Moreover, it encourages a more constructive learning experience, as students are motivated to engage with material that appropriately matches their abilities, fostering growth and development. Ultimately, the careful adjustment of exam difficulty supports the effectiveness and integrity of the assessment process.

Several human examiners showed us these difficulties and asked for our help, opening the possibilities for an exciting application of language technologies to this problem. This is why our group is quite interested in this problem and participated in this task. Actually, we are working on automatically predicting the difficulty of examinations for new language learners. The exams of our work are also in a multiple-choice format but, the number of options is lower (3 or 4, depending on the exam).

Our primary objective in this task was centered on the initiation of experiments utilizing transformer-based systems (Vaswani et al., 2017) to explore their applicability to the given problem domain. Instead of using the most modern generative models such as ChatGPT[2], Llama-2 (Touvron et al., 2023) or Mixtral (Jiang et al., 2024), we explored the use of several BERT-based models (Devlin et al., 2019), which require less computational resources. We experiment with different input sequences and use the same data and approaches for both tracks. While our results in Track 1 (Item Difficulty Prediction) were relatively low (13th position for our best run), we obtained good results in Track 2 (Response Time Prediction), where we ranked in 1st, 3rd, and 4th position with our proposed systems.

The paper is structured as follows: we describe the main features of our approach in Section 2, while we detail the runs submitted to the task in Section 3. Then, we analyzed our results in Section

[1] https://www.usmle.org/

[2] https://chat.openai.com/

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 567–570
June 20, 2024 ©2024 Association for Computational Linguistics

4. Finally, we give some conclusions and future work in Section 5.

2 Systems Description

In this Section, we describe the main features of our systems. In the development period, we tested different configurations using 10% of the training collection as test data. All our experiments are based on a BERT-base model[3] fine-tuned for regression (Devlin et al., 2019). We experimented with similar models like DeBERTa (He et al., 2021) and DistilBERT (Sanh et al., 2020), obtaining the best results with BERT. We focused on the base versions of these models instead of the large ones because we wanted to study the use of simple approaches that do not require big GPU units.

We applied the same pre-processing to all our models and focused on testing the effect of using different inputs for the model. We provide more details in the next subsections.

2.1 Pre-processing

We only used text from the item and the answers as input to our systems. More in detail, we only used the following text fields provided by the organizers:

- ItemStem_Text: contains the clinical case and the question.
- Answer_N: contains the text of the n-candidate answer.
- Answer_Text: contains the text of the correct answer,

We did not apply any special pre-processing to these input texts and used the tokenizer provided by the BERT model.

We scale the target variables (Difficulty for Track 1 and Response_Time for Track 2) into the $[0, 1]$ scale using the MinMaxScaler from sklearn[4], which gave us the best results in the development period.

2.2 Input Formats

We tested different input formats in our experiments. We wanted to explore the effects of using different combinations of text and study the importance of different text elements for solving the task. We explored the following input formats:

- **All text together**: we concatenate the ItemStem_Text field with all the Answer_N fields. With this format, we wanted to study how including all the answer candidates can help predict the difficulty of the item. We tried to include the separator token before each candidate, but we had several problems. Therefore, we just concatenated the answer candidates with the text.
- **Text and correct answer**: we include the ItemStem_Text and Answer_Text fields and use the separator token to mark the separation between the two fields. With this format, we wanted to study the impact of including only the correct answer without having access to the other answer candidates.
- **Only text**: we only include the ItemStem_Text field. We wanted to study the effect of the clinical case text, without any information about the answer candidates, when predicting the difficulty and response time.

3 Submitted Runs

We submitted the same three configurations to each Track, where the only difference between tracks is the target label used for training the models. All the runs were trained using the whole training set provided by the organizers, with the hyperparameters selected in the development period. The only difference among the three runs was the input format. All the runs were trained using a V100 GPU in Google Colab, selecting the best model after ten training epochs. The details of the three runs are:

- Run 1: it uses the ”All text together” input format described in Section 2.2. Hence, this run uses the clinical text and the candidates to make predictions. We use a batch size of 8 and a learning rate of 2e-5.
- Run 2: it uses the ”Text and correct answer” input format described in Section 2.2. This run gives us information about including the correct answer without including the other candidates. We use a batch size of 8 and a learning rate of 2e-5.
- Run 3: it uses the ”Only text” input format described in Section 2.2. The objective of this run was to make the predictions without including any information about the candidates

[3] https://huggingface.co/google-bert/bert-base-uncased

[4] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

or the correct answer. We use a batch size of 4, a learning rate of 1e-6.

Each run's batch size and learning rate were selected based on our experiments in the development period. We use the Adam optimizer and the mean-squared error as the loss function for all our experiments, while the other hyperparameters were the default provided by the transformers[5] library.

4 Analysis of Results

The official measure for both tracks was the Root Mean Squared Error metric (RMSE), which compares the prediction with the correct value. Systems are ranked according to RMSE, with the best systems obtaining the lowest error scores. We show and discuss the results of each track in the next subsections.

4.1 Track 1: Item Difficulty Prediction

In Table 1, we show the results of our three runs, the best system, and the proposed baseline in Track 1. Our best submission in this track was Run 3, which only included the clinical text as input. Thus, it seems that, at least with our approach and this data, any answer candidate's inclusion was harmful. We think this information must be helpful and want to perform a more profound study about correctly including it. Regarding the other two runs, the best one was Run 1, which included all the answer candidates.

Concerning other participants, our Run 3 was quite close to the winner system, with several systems ranking better. Besides, only Run 3 obtained better results than the proposed baseline.

Table 1: Results in Track 1, including the best system and the proposed baseline.

System	RMSE	Rank
Best system	0.299	1
Run 3	0.308	13
Baseline	0.311	16
Run 1	0.337	35
Run 2	0.363	40

4.2 Track 2: Response Time Prediction

In Table 2, we show the results of our three runs and the proposed baseline in Track 2. Our results in this task were quite good, obtaining the best result, the third and the fourth, despite using the same approaches in Track 1.

The results of the three runs were quite similar, so we must be careful with the conclusions we draw from them. According to the scores obtained, the best submission was Run 2, which included only the clinical text and the correct answer. In contrast, the submission, including the clinical text and all the answer candidates (Run 1), ranked third. Therefore, in this track, it was pretty useful to include information from the answers (in contrast to the results obtained in Track 1).

Table 2: Results in Track 1, including the best system and the proposed baseline.

System	RMSE	Rank
Run 2	23.927	1
Run 1	24.777	3
Run 3	25.365	4
Baseline	31.68	25

5 Conclusions and Future Work

Automatic prediction of exam difficulty remains an open challenge for both humans and machines. This is why the BEA 2024 Shared Task proposed evaluating systems predicting difficulty and response time in medical exams, opening a common framework for researching this challenge.

We have tested the use of BERT-based models with different input formats. Our objective was to establish a set of first results with simple systems and continue our research with the most complex approaches in the future.

We have tested the impact of using 1) only the text containing the clinical text and the question, 2) including the correct answer, and 3) including all the candidates. Our results differ depending on the track (predicting difficulty or response time). While we obtained our best results for predicting difficulty using only the clinical text, our best results for predicting response time were obtained including the correct answer.

Comparing results with other participants, we ranked at the middle of the ranking when predicting difficulty. On the other hand, we obtained the best results among all the participants when predicting response time, with our three runs in the first four positions of the final ranking.

Future work aims to study new configurations

[5] https://huggingface.co/docs/transformers/index

for both predictions and include more systems in the study.

Acknowledgements

This work has been partially funded by the Spanish Research Agency (Agencia Estatal de Investigación), DeepInfo project PID2021-127777OB-C22 (MCIU/AEI/FEDER,UE)

References

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.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Victoria Yaneva, Kai North, Peter Baldwin, An Ha Le, Saed Rezayi, Yiyun Zhou, Sagnik Ray Choudhury, Polina Harik, and Brian Clauser. 2024. Findings from the First Shared Task on Automated Prediction of Difficulty and Response Time for Multiple Choice Questions. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

The BEA 2024 Shared Task on the Multilingual Lexical Simplification Pipeline

Matthew Shardlow[1], Fernando Alva-Manchego[2], Riza Batista-Navarro[3], Stefan Bott[4], Saul Calderon Ramirez[5], Rémi Cardon[6], Thomas François[6], Akio Hayakawa[4], Andrea Horbach[7,8], Anna Hülsing[7], Yusuke Ide[9], Joseph Marvin Imperial[10,15], Adam Nohejl[9], Kai North[11], Laura Occhipinti[12], Nelson Peréz Rojas[5], Nishat Raihan[11], Tharindu Ranasinghe[13], Martin Solis Salazar[5], Sanja Štajner[14], Marcos Zampieri[11], Horacio Saggion[4]

[1]Manchester Metropolitan University [2]Cardiff University [3]University of Manchester
[4]Universitat Pompeu Fabra [5]Tecnológico de Costa Rica [6]UCLouvain
[7]University of Hildesheim [8]FernUniversität in Hagen
[9]Nara Institute of Science and Technology [10]National University Philippines
[11]George Mason University [12]University of Bologna [13]Aston University
[14]Karlsruhe [15]University of Bath
m.shardlow@mmu.ac.uk

Abstract

We report the findings of the 2024 Multilingual Lexical Simplification Pipeline shared task. We released a new dataset[1] comprising 5,927 instances of lexical complexity prediction and lexical simplification on common contexts across 10 languages, split into trial (300) and test (5,627). 10 teams participated across 2 tracks and 10 languages with 233 runs evaluated across all systems. Five teams participated in all languages for the lexical complexity prediction task and 4 teams participated in all languages for the lexical simplification task. Teams employed a range of strategies, making use of open and closed source large language models for lexical simplification, as well as feature-based approaches for lexical complexity prediction. The highest scoring team on the combined multilingual data was able to obtain a Pearson's correlation of 0.6241 and an ACC@1@Top1 of 0.3772, both demonstrating that there is still room for improvement on two difficult sub-tasks of the lexical simplification pipeline.

1 Introduction

The lexical simplification pipeline is a family of systems designed to automatically identify and replace complex vocabulary with simpler alternatives (North et al., 2023b). The lexical simplification pipeline provides a more targeted approach to simplification than automated text simplification (Al-Thanyyan and Azmi, 2021; Alva-Manchego et al., 2020; Saggion, 2017) which directly rewrites entire sentences. The two core operations included in the lexical simplification pipeline are (1) lexical complexity prediction (LCP) and (2) the replacement of complex words with simple synonyms.

LCP (Shardlow et al., 2020, 2022; North et al., 2023b,c), a form of Complex Word Identification (CWI) (Shardlow, 2013), involves assigning continuous values (0-1) to given tokens in context, representing the difficulty that an intended reader population may associate with that target word.

The second task, often referred to just as lexical simplification (LS) (Saggion et al., 2022) involves generating simple substitutions for target words in context. This task has been explored for single words and multi-word expressions, and is related to the identification of simple paraphrases (Maddela et al., 2021).

We previously identified two shortcomings of current work on the lexical simplification pipeline (Shardlow et al., 2024) as follows:

1. Current datasets only explore one pipeline operation, but no dataset exist with multiple operations on the same target words in context. This means that systems that are trained on one task are unsuitable for the other. Systems trained using multiple datasets may experience 'genre drift', where the text type across datasets differs.

2. The existing data is overwhelmingly in the English language. Whereas recent efforts exist to provide open source data in languages other than English, there is no guarantee that these datasets are created using the same protocols.

[1] https://github.com/MLSP2024/MLSP_Data/

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 571–589
June 20, 2024 ©2024 Association for Computational Linguistics

We introduce the Multilingual Lexical Simplification Pipeline (MLSP) shared task, which provides a newly annotated dataset across 10 languages for LCP and LS. The annotations for both these tasks are provided on common targets in common contexts allowing further exploration of the interplay between the two tasks and evaluation of the full pipeline on common datasets. We release data in English, Spanish, French, Portuguese, Sinhala, Filipino, Japanese, Italian, German and Catalan. Of these languages, there were previously no available LCP resources for Portuguese, Sinhala, Filipino, Italian or Catalan and no LS resources available for Sinhala, Filipino, Italian, German or Catalan.

In the remainder of this Findings paper, we overview previous related shared tasks (Section 2); give a description of the task (Section 3); overview the preparation of our shared task dataset (Section 4); the participating systems (Section 5) and the results (Section 6). We conclude with a discussion of wider factors affecting our task (Section 7).

2 Related Tasks

LS 2012 at SemEval: The first shared task in LS was proposed for SemEval 2012. It addressed English LS (Specia et al., 2012) and offered the opportunity to evaluate systems able to rank substitution candidates in relation to their simplicity. The dataset used was taken from the Lexical Substitution task at SemEval 2007 (McCarthy and Navigli, 2007) which was enriched with simplicity rankings provided by second language learners with high proficiency levels in English. The task attracted five different institutions which provided nine systems in total.

CWI 2016 at SemEval: At SemEval 2016, the CWI task (Paetzold and Specia, 2016) requested participants predict which words in a given sentence would be considered complex by a non-native English speaker. A new dataset composed of 9,200 instances was created. The task attracted 21 teams which produced a total of 42 systems. A post-completion analysis (Zampieri et al., 2017) highlighted the difficulty of the shared-task. The authors claimed that a disproportionate train/test split with over 40 times more test data, together with low inter-annotator agreement, was to blame for poor system performances.

CWI 2018 at BEA: The BEA 2018 CWI shared task (Yimam et al., 2018) proposed to tackle CWI in English, German, and Spanish (training and test data were provided), together with a multilingual task with French as a target language without training data. Teams were asked to classify words as either complex or simple (binary) and/or provide a probability for the complexity of each word. The shared task attracted eleven teams.

ALexS 2020 at IberLEF: Additionally, the IberLef 2020 forum proposed a shared task on Spanish CWI(Ortiz-Zambranoa and Montejo-Ráezb, 2020). This workshop attracted seven teams, of which three submitted to the final task. The teams competed on the newly annotated VYTEDU-CW corpus which provided binary complexity judgments over educational texts.

LCP 2021 at SemEval: The SemEval 2021 shared task on LCP (Shardlow et al., 2021) also provided a new dataset for complexity detection for single words and multi-word expressions in English attracting 55 teams. Annotations were provided as continuous complexity judgements as opposed to binary complexity values. Teams made use of deep learning based approaches to predict lexical complexity values across the corpus.

SimpleText 2021 at CLEF: The SimpleText workshop (Ermakova et al., 2022) has been running at CLEF since 2021. This workshop aims to provide benchmarks and datasets for the improvement of the accessibility of scientific information. The workshop provides datasets that participants can compete on each year in the areas of: (1) passage selection for the creation of simplified extractive summaries; (2) identification of difficult concepts and (3) query-based simplified rewriting of scientific abstracts.

TSAR 2022 Shared Task on LS: The TSAR-2022 shared task (Saggion et al., 2022) provided annotations for LS in English, Spanish and Portuguese. Participants were required to predict up to 10 simple substitutions for a complex word in each language. Participants were free to contribute to one, two or all three languages. 14 Teams submitted 60 runs across the three languages. Successful systems made use of prompt engineering (Aumiller and Gertz, 2022; Vásquez-Rodríguez et al., 2022) with large language models, as well as incorporating feature-based approaches (Li et al., 2022).

3 Task Description

Our dataset consists of instances of marked words in context, where participants are required to develop systems that first identify the complexity level of the marked word and then provide suggestions for appropriate simplifications. This unites the two previous tasks of LCP and LS into a single task, executed on common data. We have provided test data in 10 languages (Catalan, English, Filipino, French, German, Japanese, Italian, Portuguese, Sinhala, Spanish) with our final dataset totalling 300 trial instances and 5627 test instances. Participants were free to choose which tasks and language tracks they participated in.

4 Data and Resources

We initially provided participants with labelled trial data only (30 instances across 10 contexts per language, designed to indicate the format of the task). We did not provide training data, but instead pointed participants to existing resources for LCP and Complex Word Identification arising from previous shared tasks. We have provided a simplified example of the task presented to participants below:

(1) That period of **intense** regulatory **scrutiny** is a routine part of the **purchasing** process.

Token	Complexity	Substitutions
intense	0.5	strong, forceful
scrutiny	0.8	examination, observation, inspection
purchasing	0.6	buying, acquiring, obtaining

In the table above, the first column shows the tokens that were selected by the organisers for annotation. The second column shows the complexity label assigned to each word, which is provided by the participant systems. The final column shows the substitutions for each word, also provided by the participant systems. Participants provided similar annotations across their chosen language tracks, which were compared to the gold evaluation data.

4.1 Dataset Collection

Each section of the dataset was provided by a team of organisers consisting of at least one native speaker for the given language. We collected annotations from a minimum of 10 annotators per instance. Annotators were required to annotate lexical complexity for each identified token on a scale of 1-5. Annotators were also asked to provide up to 3 possible simplifications for each instance. More information on the trial dataset creation is given at Shardlow et al. (2024) and the MultiLS protocol we used at North et al. (2024).

Depending on the availability of appropriate texts requiring simplification and target populations to provide annotations, the organisers responsible for each language made autonomous decisions on the most appropriate method to gather language specific LCP and LS annotation. Information on language-specific concerns are described below.

4.1.1 Catalan

The Catalan dataset is comprised of sentences selected from the news section on education of the TeCla corpus[2] (Armengol-Estapé et al., 2021) of Catalan news texts. Target words were annotated by proficient Catalan speakers, in part recruited from persons of the social environment of the data collectors (10 participants) and in part from workers recruited via Prolific[3] crowdsourcing platform (74 participants). Although only 22% of participants were native speakers, all annotators had a high level of Catalan proficiency. The annotation process in Prolific was monitored in order to detect workers who were not following the annotation guidelines, for example, annotators who always returned the same target word as the substitute, or provided synonyms in Spanish. Non-compliant annotators were given the chance to repeat the annotation and, if they failed again, excluded.

4.1.2 English

The English dataset takes WikiBooks as a source text. English targets were identified using frequency profiling for 200 contexts. 2 additional words were identified per context ensuring that all selected words in the set were unique. The lexical complexity annotations and LS annotations were completed jointly by 21 annotators (10 native speakers, 11 non-native), all of whom were registered as students at the Manchester Metropolitan University. Each annotator saw 300 instances,

[2] https://huggingface.co/datasets/projecte-aina/tecla

[3] https://www.prolific.com/

with a total of 10-11 annotations across 600 instances.

4.1.3 Filipino

The Filipino data is composed of sentences retrieved from early-grade level books accredited by the Department of Education in the Philippines and sampled from a larger collection of Filipino resource works (Imperial and Kochmar, 2023a,b; Imperial and Ong, 2021). The genre of the sentences varies and includes samples from fiction, biographies, and instructional reference books. The annotations for the dataset were provided by 10 university staff who were native speakers of Filipino and were asked to consider the reading level of a second-grade elementary student while annotating each sentence. Instances of borrowed English words in the data were transliterated to Filipino to preserve the uniformity of phonetics (e.g. *basketball* is converted to *basketbol*).

4.1.4 French

The French dataset was compiled from a collection of texts that are used in French as a Foreign Language (FFL) classes in France, which is still under construction. The corpus contains texts targeting learners with CEFR levels going from A1 to B2. Various genres are represented, including encylopedia articles, news articles, social media, commercial and professional communication, fiction and non fiction books, or legal and political texts. Sentences that appear in the shared task dataset contain at least one word marked as B2 in the FLELex graded lexicon (François et al., 2014). Two other words were chosen manually for each sentence. The complexity annotation was performed by 10 FFL students in Belgium, attending A2 and B1 classes (5 from each level). The substitutions were provided by 10 native French speakers – Belgian master's students attending literature or social science classes.

4.1.5 German

The German data consists of Wikipedia (50%) and literary texts (50%). The data was chosen based on topics and texts mandatory for German students in their last year of secondary education in history lessons (e.g. Berlin Wall) and German lessons (e.g. *Der goldene Topf* by E. T. A. Hoffmann). Annotations were provided by German native speakers employed at universities, who were asked to take the perspective of the target group: students in their last year before graduation with a first language other than German. Simplifications that required context changes were only considered acceptable if the gender or number of a simplification required agreement with a preceding determiner, pronoun, or adjective. Example for the simplification of *Tempo*, where the determiner (underlined) changes: *mit <u>dem</u> Tempo* ("at the pace") is substituted by *mit <u>der</u> Schnelligkeit* ("at the speed").

4.1.6 Japanese

The Japanese data targets non-native Japanese speakers, whose native language is neither Chinese nor Korean, as Chinese or Korean L1 background constitutes a considerable advantage in comprehension of Japanese due to partially shared vocabulary (Koda, 1989), and therefore affects perceived lexical complexity (Ide et al., 2023).

The Japanese sentences were extracted from Wikipedia (50%), web pages with practical information, e.g. from local authorities (21%), literary fiction (19.5%), news texts (5.5%), and texts about Japanese culture and history (4%). The target words were selected to represent a wide range of word frequencies and character (*kanji*) frequencies, as well as diverse parts of speech (nouns, verbs, adjectives, adverbs, particles, and auxiliaries). Additionally, the targets include specific types of words known to be difficult for learners (compound verbs, compound particles, and onomatopoeia).

We recruited 10 non-native annotators for LCP annotation, and 10 native annotators for LS annotation. The LCP annotators were holders of Japanese Language Proficiency Test (JLPT)[4] levels 1 (N1) or 2 (N2) and their native language was neither Chinese nor Korean. The LS annotators had at least one year of experience teaching Japanese as a second language.

4.1.7 Italian

The Italian dataset comprises texts related to Italian literature, a subject taught across all school levels and grades. Specifically, 50% of the sentences have been extracted from Wikibooks, while the remaining 50% consist of sentences from 20th-century Italian authors sourced from Wikisource. We selected modern authors to avoid words considered too arcane for contemporary speakers. The task was designed for a 'general Italian speaker',

[4] https://www.jlpt.jp

and therefore, annotations were provided by native speakers with varying levels of education and literacy. A total of 215 individuals participated in the annotation process, ensuring a minimum of 10 annotations per sentence. For the substitution task, it was specified that annotators could replace target terms with words of different genders, thus not limiting the choice of possible substitutes. Additionally, annotators were instructed to treat pronominal verbs as single entities, which could also be replaced with other verbs, for example, replacing "mobilitarsi" with "agire".

4.1.8 Portuguese

The Portuguese dataset contains sentences taken from Bible extracts (47%), news articles (35%), and biomedical papers (17%). Bible instances were obtained from the Bíblia Sagrada (North et al., 2024). News instances were taken from the PorSimplesSent dataset (Leal et al., 2018) and from the CC-News (Common Crawl-News) corpus (North et al., 2022, 2023a). Biomedical instances were extracted from abstracts of biomedical literature provided by WMT-2019 (Bawden et al., 2019). Only one target word per sentence was annotated, rather the three target words per context. 21 Portuguese annotators were crowdsourced using Amazon Mechanical Turk (MTurk) and were selected from Brazil.

4.1.9 Sinhala

The Sinhala data consists of sentences extracted from a recent Sinhala news corpus (Hettiarachchi et al., 2024) and Sinhala translation of Tripitaka; the standard collection of scriptures in the Theravada Buddhist tradition written originally in Pali. Approximately 30% of the sentences were extracted from Tripitaka, and the rest of the sentences were from the news corpus. We recruited ten university students who were studying for a BA in Sinhala and were also native speakers of Sinhala for the annotation process.

4.1.10 Spanish

The Spanish dataset derives from a corpus of over 5K sentences for sentence simplification currently under development. The sentences were extracted from four online university educational books in the area of finance and were simplified following a set of simplification guidelines borrowed from the Simplext project (Saggion et al., 2015). The annotation was undertaken by 60 students who are native Spanish speakers and by 10 persons from social contacts of the data collectors, half of whom were native speakers. Out of all annotators, 8% were non-native speakers with high Spanish language proficiency.

4.2 Evaluation Metrics

For the evaluation of the LCP task we use **Pearson's correlation**, **Spearman's rank**, and the coefficient of determination (R^2) in line with the 2021 shared task on LCP.

For the evaluation of the LS task (see (Štajner et al., 2022)) we use Accuracy@k@top1 and MAP@K defined as follows the 2022 shared task on LS: **Accuracy@k@top1** is the percentage of instances where at least one of the k top-ranked substitutes matches the most frequently suggested synonym in the gold data. **MAP@k** uses a ranked list of generated substitutes, which can either be matched (relevant) or not matched (irrelevant) against the set of the gold-standard substitutes.

As some of the instances are not simplifiable or have less than k gold standard simplifications, the maximum achievable results in Accuracy@k@top1 and MAP@k are less than 1. Appendix A shows the number of unsimplifiable instances as well as maximum achievable values in all metrics.

4.3 Baselines

For LCP, we provide a baseline modelled as a linear regression on log-frequency. The frequency baseline is trained using log-frequency (minimum value if the target consists of multiple tokens) on the trial set for each language. We use frequencies provided by the `wordfreq` package[5] when possible. Additionally, since the package uses an incompatible tokenization for Japanese and does not provide any data for Sinhala, we use TUBELEX-JA[6] for Japanese, and a word frequency list for Sinhala[7] by Fernando and Dias (2021).

For LS, we provide a baseline based on zero-shot prompting a large language model. We employ the chat-finetuned Llama 2 70B model[8] (Touvron et al., 2023) in 4-bit quantisation. We use the following zero-shot prompt template and tem-

[5] https://pypi.org/project/wordfreq/

[6] https://github.com/adno/tubelex

[7] https://github.com/nlpcuom/Word-Frequency-List-for-Sinhala

[8] https://huggingface.co/meta-llama/Llama-2-70b-chat-hf

perature 0.3 to generate a maximum of 256 new tokens.

```
Context: {context}↵
Question: Given the above context, list ten
alternative {language} words for "{word}"
that are easier to understand. List only the
words without translations, transcriptions
or explanations.↵
Answer:
```

Only the ↵ symbols represent line breaks. To construct the prompt, the placeholders in curly braces are replaced by the context, the language of the instance, and the target word to be simplified. For English, the placeholder **{language}** and the subsequent space is omitted. The prompt is identical to a zero-shot prompt employed for LS using a ChatGPT model by Aumiller and Gertz (2022), except for the the underlined sentence (`List only...`), which we have added to reduce unnecessary translations to English, transcriptions to Latin alphabet, or explanations. Such extra input was generated frequently when we applied the original prompt to trial data. The addition of the sentence results in both faster inference and higher accuracy.

Our postprocessing also builds on the work by Aumiller and Gertz (2022). Based on an examination of outputs using the trial data, we made minor changes reflecting a broader array of languages and scripts as well as a different model. For instance, we allow words to be separated by ideographic commas (、) commonly used in Japanese, or lists enumerated using letters (e.g. `a)`, `b)`, ...), which occurred in Llama 2 output.

5 Participating Systems

ANU (Seneviratne and Suominen, 2024) The ANU team relied on a prompting strategy with GPT-3.5 (i.e. GPT-3.5-turbo-instruct) for both tasks using zero, one, and few-shot strategies. The zero-shot strategy included the context and target word while the non-zero strategies relied on instructing the model with one or three random samples from the trial data according to the prompting template. For LS, a combination of filtering and substitution was applied. Overall, the authors indicate under-performance for the LCP task while strong performance for English in LS.

Archaeology (Cristea and Nisioi, 2024) The Archaeology team participated in both LCP and LS. For both tasks, they make use of machine translation software to convert all texts to English. The LCP values are generated using a feature-based approach with word-level, syntactic-level and semantic-level features. An XGBoost regressor is trained on the Semeval 2021 English test dataset and used to predict lexical complexity values for all languages. The simplifications for the LS task were generated in English using the translated data by prompting a large language model (OpenHermes 2.5) to produce JSON data containing the candidate replacements and back-translated to the target language.

CocoNut The CocoNut team submitted LAE-LS, which introduced a novel method for LS, trained without the use of parallel corpora or external linguistic resources. LAE-LS employed an Adversarial Editing System with guidance from a confusion loss and an invariance loss to predict lexical edits in the original sentences. An LLM-enhanced loss was tailored to distill high-quality knowledge from LLMs into the Edit Predictor. Complex words within sentences were masked and a Difficulty-aware Filling module crafted to replace masked positions with simpler words. For LCP, the team used the probability of a word being masked by the Edit Predictor as the complexity value of the word in context. For LS, complex words were masked and the Difficulty-aware Filling module was used to predict substitute words.

GMU (Goswami et al., 2024) The GMU team participated in both subtasks. For LCP they employed a weighted ensemble of mBERT, XLM-R and language specific BERT models. All trial data was used for cross-lingual training and evaluation. For the combined track, an ensemble of language specifc models was used. For LS GPT4-turbo zero shot prompting was used, as well as mBERT, XLM-R and language specific BERT models. Cosine similarity between the target token and the substitutions generated by all the models were generated. Sentence transformer LaBSE is used to find the embeddings of the substitutions. The top 10 substitutions with the highest cosine similarity are selected for the output.

ISEP Presidency University (Dutilleul et al., 2024) The ISEP team also relied on a GPT-3 language model (i.e. GPT-3.5-turbo-instruct) and prompt engineering to solve the LS task. More concretely, several prompt generation strategies are used: a context-free strategy asks for ten simpler substitutes for the target word without specifying the context, a zero-shot strategy instead provides the context and the target word, a one-shot strategy is similar to zero-shot but provides one example of how to answer, and finally a few-shots strategy provide several examples to the model before testing. Responses from all strategies are aggregated and answers ranked to produce the final list of substitutes. The team reports satisfactory aggregated performance in most languages they applied this method to.

ITEC (Tack, 2024) The ITEC team participated only in the LCP subtask for French. They relied on two pre-trained models, previously developed for personalised LCP. Due to the characteristics of the shared task data, the personalisation component was removed. The team employed two models of similar architectures: a mix of character and FastText embeddings that are fed to either a BiLSTM or a feed-forward network, in order to consider contextual information or not, respectively, for predictions.

RETUYT-INCO (Sastre et al., 2024) The RETUYT-INCO team make use of a range of methods for their submitted runs, including word embeddings and frequency baselines for Spanish, English and Portuguese (LS). Feed forward networks with BERT-based embeddings for Spanish and English (LCP). Fine-tuning Mistral-7B for English (LCP) and with synthetic data and self-consistency for English, Spanish, Catalan and Portuguese (LCP and LS) and finally, prompting strategies using models available in the Groq API for Spanish (LS).

SCaLAR The SCaLAR team participated across both tasks, employing Mistral-7B for LS in a few shot learning setup with post-processing. Similarity scores were obtained through Word2Vec to identify the the top 10 similar words for each complex word. For LCP, the team used a weighted sum of 2 approaches: (1) MPNet Hidden State to Image Regression with EfficientNet: Transforms MPNet hidden states into image format and employs EfficientNet for image regression, bridging text data to convolutional neural networks. (2) XGBoost Regressor with TF-IDF and Zipf Frequency Features: Utilizes XGBoost regressor with features derived from TF-IDF and Zipf frequency.

SDJZUandUU The Complex Word Identification (CWI) model of team SDJZUandUU comprises of three integral modules: the Feature Collection Module, Feature Fusion Module, and Regression Model. The Feature Collection Module is designed to gather diverse feature sets including 16 commonly utilized handcrafted features, GloVe embeddings, and dynamic dependency embeddings. This module incorporates Gaussian vectorization techniques to vectorize the handcrafted features effectively. Subsequently, the Feature Fusion Module combines the aforementioned feature types into a vector representation, which is then passed to the Regression Model. The Regression Model is composed of three layers: two Support Vector Regression (SVR) polynomial layers for feature refinement within the feature vectors, and one feedforward layer aimed at predicting the final complexity value.

TMU-HIT (Enomoto et al., 2024) TMU-HIT employed a GPT-4 based approach in both tasks. In System 1, the team used GPT-4 to generate 10 alternative words for the target word in a zero-shot setting. In the case of Japanese, rather than solely generating alternative words, the team directed GPT-4 to generate sentences wherein the target words were substituted with each alternative word. This approach was necessary to ensure that the “katsuyou” (inflection) appropriately suited the context in Japanese. substitues were reranked through (a) prompting and (b) fine-tuned XGLM. For LCP, the team use a chain-of-thought based prompting method employing GPT-4 to generate an instruction in English, and subsequently assigning complexity scores to target words across all languages based on the English instruction.

6 Results

The full results for LCP and LS are displayed in Appendix B and Appendix C respectively. Each team was permitted to submit up to 3 runs per language track, with teams permitted to submit to both the combined track and the individual language tracks. The ID field indicates the run ID of the participants systems. Where teams submitted a separate system to the combined track, the results for each individual language were also separately processed and included in the results tables for the individual language tracks, these are indicated by a run ID preceding with ‘A’. All team outputs can be found via GitHub[9].

Whilst all systems provide interesting insights into the nature of the lexical simplification pipeline, we have chosen to highlight a small number of systems below. The full descriptions of each system are available in the proceedings.

The results demonstrate that the GPT-4 based approach of the *TMU-HIT* team performed well across both tasks and all language tracks. This system consistently outperforms the baseline and is consistently the first or second highest ranked system. Prompt-based strategies have previously proved to be effective for LS, but not for the LCP task.

The *Archaeology* submission based on machine translation performs well for LCP, ranking as the second team in the combined track. This system uses a feature-based regression, demonstrating that this is still a competitive approach. The system does not perform as well on the LS task, and this is likely due to the challenge of correctly identifying targets after back-translation.

The *RETUYT-INCO* submission attains second place in LCP for Catalan, Filipino, Sinhala and Spanish. This submission made use of bespoke resources, including synthetic data for low-resource languages. The competitive performance of this submission on these tracks indicates that this approach may be appropriate for future low-resource languages that cannot be handled through a conventional prompt-based approach.

The *GMU* team attained first place for the EN-LCP task, setting a new hard to beat baseline for this dataset. Their approach also attains strong LS results for all languages, consistently attaining the 2nd or third ranked team in each language and ranking as the second team on the combined track.

Finally, the *ISEP* team chose to only compete in a reduced set of languages for the LS task. This focus allowed them to submit a competitive system for Catalan (1st place), Portuguese (1st place), French (2nd team) as well as English (4th Team) and German (4th team), outperforming the baseline in all cases.

We provided a simple baseline for LCP based on word-frequency and for LS based on a simple LLM-prompting strategy following prior work. The baseline is included in all results tables as 'Baseline', except for the combined results table, where we have not included a baseline result. We have sorted each results table, including the baselines, according to the Pearson's Correlation for LCP and Acc@1@Top1 for LS and we refer to systems 'above the baseline' in this context.

For LCP our baseline system was generally competitive, expect for Sinhala. The system was based on word frequencies and the frequencies we had available for Sinhala were not suitable for the task. Our baseline received a negative correlation to the gold labels for Sinhala (as did several participant systems). For other systems, our baseline performs strongly (ranking between the 2nd and 4th system for all languages except for English and Sinhala) confirming our hypothesis that word frequency would be a strong indicator of lexical difficulty. For English, the baseline system attains a strong correlation of 0.7480, but is outperformed by 9 other systems. The English LCP track was more subscribed than any other.

For LS, our baseline system received mixed results, generally attaining a mid-table ranking. Our approach was to reuse the prompt from the previous LS shared-task winner, which is a similar strategy to many of the submitted systems which also further improved on this same approach. Our system performs particularly poorly for Filipino and for Sinhala, and this is likely the result of the base language model lacking training data for these languages.

Although we have ranked our systems according to Pearson's correlation for LCP, it is also interesting to observe the R^2 metric of each system as compared to the baseline. The R^2 metric describes the proportion of variance captured by the system's results, i.e., how well do the LCP values returned by the system describe the LCP values in the gold labels. A negative R^2 indicates that the returned values are a poor fit to the gold values, whereas a positive R^2 indicates a good fit.

Our baseline attains a positive R^2 for all systems, except for Catalan and Sinhala. Notably, for English the baseline system attains the highest R^2 of any system. This is also true for Filipino (all other systems have negative R^2), German and Portuguese. This indicates that although systems are able to provide correlative LCP judgements, additional factors are still required to fully represent the underlying data distributions.

[9] https://github.com/MLSP2024/MLSP_Participants/

7 Discussion

We provided 10 languages for the evaluation of LCP and LS. Unsurprisingly, the most subscribed language track was English, with the most prior work and existing resources in NLP concentrated on English. We hope to address the imbalance in LCP/LS research by providing equal amounts of data for all languages that we have included. The English submissions attained the highest scores overall for LCP and LS, demonstrating that the English task is better resourced. Further developments in multilingual NLP and in bespoke resources for individual target languages will help to improve the performance of other systems on the tasks in our dataset.

Our dataset covers widespread global languages such as English, Spanish and French. There are a disproportionate number of languages in our dataset that are influenced from the romance family (Spanish, Catalan, French, Italian, Portuguese). We hope to extend the dataset in further iterations to include other widespread languages such as Mandarin Chinese, Hindi, Modern Standard Arabic and Bengali.

In addition to focussing future development on widespread languages, our work has also shown that LCP and LS can be effectively applied to low-resource languages. Future work to develop LCP/LS resources using the MultiLS framework (North et al., 2024) which we have followed will be incorporated into our dataset to enable the LS task for wider digital communities.

Whereas previous approaches to LCP have focussed on regression studies, e.g., using a language model with a regression head, it is interesting to note that many of the systems were able to use a prompting strategy to get good results for the LCP task. The TMU-HIT system relies on prompting to generate N judgements, effectively forcing the LM to undertake the annotation task. This proves effective across many languages. The use of language models to replicate the annotators is an interesting area of future exploration which may have significant repercussions across other similar lexical semantics tasks such as hate speech or sentiment analysis. Nonetheless, feature based systems such as the frequency baseline and the feature-based regression of the Archaeology team still performed competitively, demonstrating that this can be an effective method for LCP, especially when large language models are not available for the target language.

The principal strategy for the LS task employed by our participants was through prompt engineering. It is worth noting that several of the top-ranked submissions on this task used GPT4/GPT3.5, both of which are closed-source proprietary models. Whilst differing prompt engineering strategies were employed throughout the task, it is very difficult to separate the differences in performances that can be attributed to (a) the prompting strategies used and (b) the language models that they have been applied to. A possible future strategy to prevent model-variance may be to provide all teams access to some common model and enforce its use in a task.

8 Conclusion

We present the findings of the 2024 Multilingual Lexical Simplification Pipeline shared task hosted at the 19th Workshop on Innovative Use of NLP for Building Educational Applications. We provided the first multilingual dataset for LCP and LS on common targets, spanning ten languages and nearly 6,000 instances. Ten teams participated in our task employing a range of LLM-based strategies at the forefront of modern NLP. Seven teams submitted system description papers. Our shared task has progressed the forefront of lexical simplification research and the organisers look forward to seeing future multilingual lexical simplification research born of these efforts. All datasets, baselines and participant submissions are available through the MLSP2024 GitHub Organisation[10].

Acknowledgments

Joseph Imperial is supported by the National University Philippines (Project ID: 2023I-1T-05-MLA-CCIT-Computer Science) and the UKRI Centre for Doctoral Training in Accountable, Responsible and Transparent AI [EP/S023437/1] of the University of Bath.

Andrea Horbach's work is part of the research conducted at CATALPA – Center for Advanced Technology-Assisted Learning and Predictive Analytics of the FernUniversität in Hagen, Germany. Anna Hülsing is supported by the German Federal Ministry of Education and Research (grant no. FKZ 01JA23S03C).

[10] https://github.com/MLSP2024

The Japanese section of the dataset was created as a part a joint research project of Nara Institute of Science and Technology and Nikkei, Inc.

Horacio Saggion, Stefan Bott and Akio Hayakawa acknowledge funding from the European Uniont's Horizon Europe research and innovation program under the Grant Agreement No. 101132431 (iDEM Project) – views and opinions expressed are however those of the author(s) only and do necessarily reflect those of the European Union. Horacio Saggion, Stefan Bott and Akio Hayakawa also thank the support from Departament de Recerca i Universitats de la Generalitat de Catalunya (ajuts SGR-Cat 2021).

The English section of the dataset was annotated thanks to the Manchester Metropolitan University RISE programme. The organisers would like to extend their thanks to the organisers of the RISE programme and to the student annotators who participated in the annotation process for their timely completion and attendance at annotation meetings.

The organisers would like to thank all the annotators that participated across the ten languages we have provided, both paid and unpaid. This work would not have been possible without the kind support of friends and professional connections within the organiser's networks.

We would also like to thank the ten teams, consisting of 37 colleagues at various levels of their academic journeys, for their participation in the shared task. Particular thanks go to those that provided reviews for other participants shared task submissions.

Finally, we would like to thank the organisers of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024), who kindly hosted our shared task. Particular thanks go to Victoria Yaneva for serving as liaison to the BEA organisers and also to Andrea Horbach for her help in facilitating the Softconf track.

References

Suha S. Al-Thanyyan and Aqil M. Azmi. 2021. Automated text simplification: A survey. *ACM Comput. Surv.*, 54(2).

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2020. Data-driven sentence simplification: Survey and benchmark. *Computational Linguistics*, 46(1):135–187.

Jordi Armengol-Estapé, Casimiro Pio Carrino, Carlos Rodriguez-Penagos, Ona de Gibert Bonet, Carme Armentano-Oller, Aitor Gonzalez-Agirre, Maite Melero, and Marta Villegas. 2021. Are multilingual models the best choice for moderately under-resourced languages? A comprehensive assessment for Catalan. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4933–4946, Online. Association for Computational Linguistics.

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.

Rachel Bawden, Kevin Bretonnel Cohen, Cristian Grozea, Antonio Jimeno Yepes, Madeleine Kittner, Martin Krallinger, Nancy Mah, Aurelie Neveol, Mariana Neves, Felipe Soares, Amy Siu, Karin Verspoor, and Maika Vicente Navarro. 2019. Findings of the WMT 2019 biomedical translation shared task: Evaluation for MEDLINE abstracts and biomedical terminologies. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 29–53, Florence, Italy. Association for Computational Linguistics.

Petru Theodor Cristea and Sergiu Nisioi. 2024. Archaeology at MLSP 2024: Machine translation for lexical complexity prediction and lexical simplification. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Benjamin Dutilleul, Mathis Debaillon, and Sandeep Mathias. 2024. ISEP presidency university at MLSP 2024: Using GPT-3.5 to generate substitutes for lexical simplification. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Taisei Enomoto, Hwichan Kim, Tosho Hirasawa, Yoshinari Nagai, Ayako Sato, Kyotaro Nakajima, and Mamoru Komachi. 2024. TMU-HIT at MLSP 2024: How well can GPT-4 tackle multilingual lexical simplification? In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Liana Ermakova, Patrice Bellot, Jaap Kamps, Diana Nurbakova, Irina Ovchinnikova, Eric SanJuan, Elise Mathurin, Sílvia Araújo, Radia Hannachi, Stéphane Huet, et al. 2022. Automatic simplification of scientific texts: Simpletext lab at clef-2022. In *European Conference on Information Retrieval*, pages 364–373. Springer.

Aloka Fernando and Gihan Dias. 2021. Building a linguistic resource : A word frequency list for Sinhala.

In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, pages 606–610, National Institute of Technology Silchar, Silchar, India. NLP Association of India (NLPAI).

Thomas François, Núria Gala, Patrick Watrin, and Cédrick Fairon. 2014. Flelex: a graded lexical resource for french foreign learners. In *International conference on Language Resources and Evaluation (LREC 2014)*.

Dhiman Goswami, Kai North, and Marcos Zampieri. 2024. GMU at MLSP 2024: Multilingual lexical simplification. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Hansi Hettiarachchi, Damith Premasiri, Lasitha Uyangodage, and Tharindu Ranasinghe. 2024. NSINA: A News Corpus for Sinhala. In *The 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*.

Yusuke Ide, Masato Mita, Adam Nohejl, Hiroki Ouchi, and Taro Watanabe. 2023. Japanese lexical complexity for non-native readers: A new dataset. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 477–487, Toronto, Canada. Association for Computational Linguistics.

Joseph Marvin Imperial and Ekaterina Kochmar. 2023a. Automatic readability assessment for closely related languages. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5371–5386, Toronto, Canada. Association for Computational Linguistics.

Joseph Marvin Imperial and Ekaterina Kochmar. 2023b. BasahaCorpus: An expanded linguistic resource for readability assessment in Central Philippine languages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6302–6309, Singapore. Association for Computational Linguistics.

Joseph Marvin Imperial and Ethel Ong. 2021. Under the microscope: Interpreting readability assessment models for Filipino. In *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation*, pages 1–10, Shanghai, China. Association for Computational Lingustics.

Keiko Koda. 1989. The effects of transferred vocabulary knowledge on the development of L2 reading proficiency. *Foreign Lang. Ann.*, 22(6):529–540.

Sidney Evaldo Leal, Magali Sanches Duran, and Sandra Maria Aluísio. 2018. A nontrivial sentence corpus for the task of sentence readability assessment in Portuguese. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 401–413, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Xiaofei Li, Daniel Wiechmann, Yu Qiao, and Elma Kerz. 2022. MANTIS at TSAR-2022 shared task: Improved unsupervised lexical simplification with pretrained encoders. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 243–250, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Mounica Maddela, Fernando Alva-Manchego, and Wei Xu. 2021. Controllable text simplification with explicit paraphrasing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3536–3553, Online. Association for Computational Linguistics.

Diana McCarthy and Roberto Navigli. 2007. SemEval-2007 task 10: English lexical substitution task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 48–53, Prague, Czech Republic. Association for Computational Linguistics.

Kai North, Alphaeus Dmonte, Tharindu Ranasinghe, Matthew Shardlow, and Marcos Zampieri. 2023a. ALEXSIS+: Improving substitute generation and selection for lexical simplification with information retrieval. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 404–413, Toronto, Canada. Association for Computational Linguistics.

Kai North, Tharindu Ranasinghe, Matthew Shardlow, and Marcos Zampieri. 2023b. Deep Learning Approaches to Lexical Simplification: A Survey. *Preprint*, arXiv:2305.12000.

Kai North, Tharindu Ranasinghe, Matthew Shardlow, and Marcos Zampieri. 2024. Multils: A multi-task lexical simplification framework. *Preprint*, arXiv:2402.14972.

Kai North, Marcos Zampieri, and Tharindu Ranasinghe. 2022. ALEXSIS-PT: A new resource for Portuguese lexical simplification. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6057–6062, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Kai North, Marcos Zampieri, and Matthew Shardlow. 2023c. Lexical complexity prediction: An overview. *ACM Computing Surveys*, 55(9):1–42.

Jenny A Ortiz-Zambranoa and Arturo Montejo-Ráezb. 2020. Overview of alexs 2020: First workshop on lexical analysis at sepln. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020)*, volume 2664, pages 1–6.

Gustavo Paetzold and Lucia Specia. 2016. SemEval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569, San Diego, California. Association for Computational Linguistics.

Horacio Saggion. 2017. *Automatic Text Simplification*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Horacio Saggion, Sanja Štajner, Stefan Bott, Simon Mille, Luz Rello, and Biljana Drndarevic. 2015. Making it Simplext: Implementation and Evaluation of a Text Simplification System for Spanish. *TACCESS*, 6(4):14.

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.

Ignacio Sastre, Leandro Alfonso, Facundo Fleitas, Federico Gil, Andrés Lucas, Tomás Spoturno, Santiago Góngora, Aiala Rosá, and Luis Chiruzzo. 2024. RETUYT-INCO at MLSP 2024: Experiments on language simplification using embeddings, classifiers and large language models. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Sandaru Seneviratne and Hanna Suominen. 2024. ANU at MLSP 2024: Prompt-based lexical simplification for english and sinhala. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Matthew Shardlow. 2013. A comparison of techniques to automatically identify complex words. In *51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop*, pages 103–109, Sofia, Bulgaria. Association for Computational Linguistics.

Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Huelsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Peréz Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, Marcos Zampieri, and Horacio Saggion. 2024. An Extensible Massively Multilingual Lexical Simplification Pipeline Dataset using the MultiLS Framework. In *Proceedings of the 3rd Workshop on Tools and Resources for People with REAding DIfficulties (READI)*.

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.

Lucia Specia, Sujay Kumar Jauhar, and Rada Mihalcea. 2012. SemEval-2012 task 1: English lexical simplification. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 347–355, Montréal, Canada. Association for Computational Linguistics.

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*, 5:991242.

Anaïs Tack. 2024. ITEC at MLSP 2024: Transferring predictions of lexical difficulty from non-native readers. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *ArXiv preprint*, arXiv:2307.09288 [cs].

Laura Vásquez-Rodríguez, Nhung Nguyen, Matthew Shardlow, and Sophia Ananiadou. 2022. UoM&MMU at TSAR-2022 shared task: Prompt learning for lexical simplification. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 218–224, Abu Dhabi, United Arab Emirates (Virtual). 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.

Marcos Zampieri, Shervin Malmasi, Gustavo Paetzold, and Lucia Specia. 2017. Complex Word Identification: Challenges in Data Annotation and System Performance. In *Proceedings of the 4th Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA 2017)*, Taipei, Taiwan. Asian Federation of Natural Language Processing.

A Dataset Statistics and Maximum Achievable Results

Language	# Test Instances	# Unsimplifiable	Max. MAP@1, Accuracy@k@Top1	Max. MAP@3	Max. MAP@5
All	5627	133	0.9763	0.9081	0.7963
Catalan	445	1	0.9977	0.9910	0.9793
English	570	0	1.0000	0.9491	0.8115
Filipino	570	130	0.7719	0.5222	0.3466
French	570	0	1.0000	0.9953	0.9673
German	570	0	1.0000	0.9309	0.7908
Italian	570	0	1.0000	0.9859	0.9228
Japanese	570	0	1.0000	0.9988	0.9957
Portuguese	568	1	0.9982	0.9241	0.7220
Sinhala	600	0	1.0000	0.8072	0.4873
Spanish	593	1	0.9983	0.9966	0.9885

B Lexical Complexity Prediction Results

Team Name	Language	ID	Pearson’s	Spearman’s	R^2
TMU-HIT	All	2	0.6241	0.6215	0.2456
TMU-HIT	All	1	0.5609	0.5697	-0.3111
Archaeology	All	2	0.5316	0.5415	0.2560
RETUYT-INCO	All	1	0.4858	0.4892	-0.6746
GMU	All	1	0.3494	0.3642	0.1094
SCaLAR	All	1	0.0979	-0.0104	-0.0301

Team Name	Language	ID	Pearson’s	Spearman’s	R^2
TMU-HIT	Catalan	A2	0.6158	0.5989	-0.1610
TMU-HIT	Catalan	A1	0.5279	0.5327	-0.9634
RETUYT-INCO	Catalan	1	0.3948	0.3862	-1.3972
RETUYT-INCO	Catalan	A1	0.3608	0.3564	-1.5394
Baseline	Catalan	1	0.3011	0.3106	-0.3698
Archaeology	Catalan	1	0.2960	0.3029	-0.0342
Archaeology	Catalan	2	0.2744	0.2649	0.0110
GMU	Catalan	1	0.1549	0.1574	-0.3378
GMU	Catalan	A1	0.1137	0.1081	-0.1453
SCaLAR	Catalan	A1	0.0424	0.0065	-0.2236

Team Name	Language	ID	Pearson’s	Spearman’s	R^2
GMU	English	1	0.8497	0.7984	0.5247
TMU-HIT	English	2	0.8198	0.7552	0.5147
SDJZUandUU	English	3	0.8123	0.7754	0.5245
SDJZUandUU	English	1	0.8111	0.7414	0.3731
RETUYT-INCO	English	1	0.8061	0.7596	0.3154
TMU-HIT	English	1	0.8036	0.7017	0.3161
Archaeology	English	2	0.7904	0.7547	0.4393
SDJZUandUU	English	2	0.7820	0.7182	0.3529
RETUYT-INCO	English	3	0.7599	0.7406	-0.1796
Baseline	English	1	0.7480	0.7451	0.5475
RETUYT-INCO	English	2	0.5502	0.4923	0.1062
ANU	English	1	0.3358	0.3591	-3.0241
GMU	English	A1	0.3118	0.3183	0.0585
CocoNut	English	1	0.1972	0.2160	-5.1596
ANU	English	3	0.1915	0.2402	-0.5842
ANU	English	2	0.1789	0.2285	-0.0917
SCaLAR	English	A1	0.0126	0.0139	-0.2984

Team Name	Language	ID	Pearson's	Spearman's	R^2
TMU-HIT	Filipino	A1	0.5692	0.5816	-0.3536
TMU-HIT	Filipino	A2	0.5013	0.5244	-2.4778
RETUYT-INCO	Filipino	A1	0.4640	0.4540	-1.4847
Archaeology	Filipino	2	0.4427	0.4476	-0.0763
Baseline	Filipino	1	0.3892	0.4178	0.0036
Archaeology	Filipino	1	0.3620	0.4133	-0.9131
GMU	Filipino	A1	0.2823	0.2767	-0.0457
GMU	Filipino	1	0.1942	0.1908	-0.0824
SCaLAR	Filipino	A1	-0.0700	-0.0792	-0.2649

Team Name	Language	ID	Pearson's	Spearman's	R^2
TMU-HIT	French	A1	0.6253	0.6302	0.2704
Archaeology	French	1	0.5335	0.5310	0.2136
TMU-HIT	French	A2	0.5278	0.5343	0.2391
Baseline	French	1	0.5166	0.5221	0.1458
RETUYT-INCO	French	A1	0.4868	0.4651	0.0279
Archaeology	French	2	0.4411	0.4188	0.1862
ITEC	French	2	0.3607	0.4972	-4.4459
ITEC	French	1	0.3253	0.3533	-3.3488
GMU	French	1	0.3193	0.3207	0.0484
GMU	French	A1	0.1557	0.1756	0.0039
SCaLAR	French	A1	0.1035	0.0674	0.0061

Team Name	Language	ID	Pearson's	Spearman's	R^2
TMU-HIT	German	A2	0.7177	0.7365	-0.5585
TMU-HIT	German	A1	0.6582	0.6813	-0.7654
Baseline	German	1	0.5912	0.6096	0.0727
Archaeology	German	2	0.5577	0.5774	-0.1320
Archaeology	German	1	0.5508	0.5726	0.0686
RETUYT-INCO	German	A1	0.3909	0.3981	-0.3463
GMU	German	A1	0.1402	0.1473	-0.5279
SCaLAR	German	A1	0.0310	0.0177	-1.2467
GMU	German	1	0.0123	0.0095	-1.1301

Team Name	Language	ID	Pearson's	Spearman's	R^2
TMU-HIT	Italian	A2	0.6011	0.6220	0.2425
TMU-HIT	Italian	A1	0.5391	0.5557	-1.7874
Archaeology	Italian	1	0.5341	0.5320	-0.4175
Baseline	Italian	1	0.5186	0.5417	0.2265
RETUYT-INCO	Italian	A	0.4945	0.5128	-2.6399
Archaeology	Italian	2	0.4790	0.4805	-0.0599
GMU	Italian	1	0.2919	0.2961	0.0770
GMU	Italian	A1	0.1797	0.1706	-0.0064
SCaLAR	Italian	A1	-0.0234	-0.0425	-0.0643

Team Name	Language	ID	Pearson's	Spearman's	R^2
TMU-HIT	Japanese	2	0.7333	0.7305	0.4129
TMU-HIT	Japanese	1	0.6448	0.6479	-0.0958
Baseline	Japanese	1	0.6420	0.6684	0.3395
Archaeology	Japanese	2	0.4851	0.5126	-0.0983
RETUYT-INCO	Japanese	A1	0.4054	0.4073	-0.5215
Archaeology	Japanese	1	0.2803	0.2648	-2.2358
GMU	Japanese	A1	0.1775	0.1827	0.0241
GMU	Japanese	1	0.0350	0.0408	-0.0393
SCaLAR	Japanese	A1	-0.0660	-0.0784	-0.1007

Team Name	Language	ID	Pearson's	Spearman's	R^2
TMU-HIT	Portuguese	A2	0.7858	0.7988	0.1533
TMU-HIT	Portuguese	A1	0.7638	0.7729	-0.4987
Baseline	Portuguese	1	0.7126	0.7427	0.4890
Archaeology	Portuguese	1	0.7143	0.7102	-0.2612
Archaeology	Portuguese	2	0.6831	0.6923	0.2419
RETUYT-INCO	Portuguese	1	0.6772	0.7121	-1.5487
RETUYT-INCO	Portuguese	A1	0.6571	0.6899	-1.5931
SCaLAR	Portuguese	A1	0.0490	0.0270	-0.1825

Team Name	**Language**	**ID**	**Pearson's**	**Spearman's**	R^2
TMU-HIT	Sinhala	A2	0.3081	0.3343	-1.6030
TMU-HIT	Sinhala	A1	0.2482	0.3261	-3.0794
RETUYT-INCO	Sinhala	A1	0.1344	0.1094	-7.2755
GMU	Sinhala	1	0.1246	0.1303	-0.0370
ANU	Sinhala	2	0.0534	0.0866	-2.3263
SCaLAR	Sinhala	A1	0.0450	0.0279	-0.9819
Archaeology	Sinhala	2	0.0437	0.0298	-0.4590
GMU	Sinhala	A1	0.0263	0.0284	-0.1142
ANU	Sinhala	1	-0.0108	-0.0105	-15.5689
ANU	Sinhala	3	-0.0162	0.0487	-1.5636
Archaeology	Sinhala	1	-0.0290	-0.0272	-9.3516
Baseline	Sinhala	1	-0.1955	-0.2564	-0.2875

Team Name	**Language**	**ID**	**Pearson's**	**Spearman's**	R^2
TMU-HIT	Spanish	A2	0.7616	0.7460	0.4940
TMU-HIT	Spanish	A1	0.7201	0.6796	-0.0991
RETUYT-INCO	Spanish	2	0.6641	0.6547	0.2808
RETUYT-INCO	Spanish	A1	0.6397	0.6296	0.2541
Baseline	Spanish	1	0.5513	0.5299	0.2556
Archaeology	Spanish	1	0.5274	0.4793	0.2507
Archaeology	Spanish	2	0.5034	0.4588	0.2304
RETUYT-INCO	Spanish	1	0.3126	0.2369	0.0131
GMU	Spanish	1	0.2438	0.1984	-0.0731
GMU	Spanish	A1	0.1957	0.1772	-0.0806
SCaLAR	Spanish	A1	-0.0009	0.0180	-0.0367

C Lexical Simplification Results

Team Name	Language	ID	Acc@1@Top1	Acc@3@Top1	MAP@3	MAP@5
TMU-HIT	All	1	0.3772	0.5498	0.4652	0.3421
TMU-HIT	All	2	0.3573	0.5498	0.457	0.3371
GMU	All	1	0.3345	0.4828	0.379	0.2754
TMU-HIT	All	3	0.2933	0.5498	0.4461	0.3306
RETUYT-INCO	All	1	0.2156	0.3324	0.2412	0.165
RETUYT-INCO	All	2	0.2074	0.3216	0.2351	0.1608
GMU	All	2	0.1331	0.2999	0.1981	0.1561
Archaeology	All	A1	0.0538	0.134	0.0882	0.0713

Team Name	Language	ID	Acc@1@Top1	Acc@3@Top1	MAP@3	MAP@5
ISEP	Catalan	1	0.2719	0.3932	0.5003	0.3759
TMU-HIT	Catalan	A1	0.2584	0.3707	0.469	0.3547
TMU-HIT	Catalan	A2	0.2516	0.3707	0.4578	0.348
GMU	Catalan	1	0.2247	0.328	0.362	0.2641
RETUYT-INCO	Catalan	A1	0.1977	0.2943	0.3024	0.21
Baseline	Catalan	1	0.1977	0.2898	0.3000	0.2121
TMU-HIT	Catalan	A3	0.1955	0.3707	0.4528	0.345
RETUYT-INCO	Catalan	A2	0.1932	0.2831	0.3077	0.2106
GMU	Catalan	2	0.0651	0.1595	0.172	0.1408
Archaeology	Catalan	2	0.0404	0.1101	0.1203	0.0972
Archaeology	Catalan	1	0.0292	0.0651	0.069	0.0556

Team Name	Language	ID	Acc@1@Top1	Acc@3@Top1	MAP@3	MAP@5
TMU-HIT	English	1	0.5245	0.7456	0.5762	0.4142
GMU	English	1	0.5157	0.6894	0.513	0.3691
ANU	English	3	0.5105	0.6649	0.5324	0.3744
ANU	English	1	0.4684	0.6561	0.5069	0.3652
ISEP	English	1	0.4684	0.6754	0.5351	0.3877
ANU	English	2	0.4631	0.6421	0.4978	0.3524
TMU-HIT	English	2	0.4438	0.7456	0.5595	0.4042
Baseline	English	1	0.3877	0.5631	0.4241	0.2956
RETUYT-INCO	English	3	0.3789	0.5701	0.3832	0.2634
RETUYT-INCO	English	2	0.3438	0.5526	0.3718	0.2542
CocoNut	English	1	0.2298	0.3877	0.2303	0.1674
GMU	English	A2	0.1929	0.4157	0.2339	0.1869
GMU	English	2	0.1859	0.3561	0.1945	0.1454
Archaeology	English	2	0.0947	0.2578	0.151	0.1272

Team Name	Language	ID	Acc@1@Top1	Acc@3@Top1	MAP@3	MAP@5
TMU-HIT	Filipino	A1	0.065	0.0878	0.1807	0.1189
TMU-HIT	Filipino	A2	0.0615	0.0878	0.1736	0.1147
GMU	Filipino	A1	0.0562	0.0685	0.1395	0.0916
GMU	Filipino	1	0.0561	0.0684	0.1392	0.0914
TMU-HIT	Filipino	A3	0.0404	0.0878	0.1592	0.1061
Archaeology	Filipino	1	0.0175	0.0298	0.0313	0.0215
GMU	Filipino	2	0.0157	0.0245	0.0449	0.0338
RETUYT-INCO	Filipino	A1	0.0087	0.0087	0.0154	0.0094
Archaeology	Filipino	2	0.007	0.0122	0.0141	0.0095
RETUYT-INCO	Filipino	A2	0.007	0.0087	0.0082	0.0051
Baseline	Filipino	1	0.007	0.007	0.0225	0.014

Team Name	Language	ID	Acc@1@Top1	Acc@3@Top1	MAP@3	MAP@5
TMU-HIT	French	A1	0.426	0.6197	0.6977	0.5466
TMU-HIT	French	A2	0.4242	0.6197	0.694	0.5443
ISEP	French	1	0.3743	0.5711	0.6484	0.4996
GMU	French	A1	0.3661	0.514	0.5148	0.3946
GMU	French	1	0.3655	0.5131	0.5141	0.394
TMU-HIT	French	A3	0.3257	0.6197	0.6815	0.5368
RETUYT-INCO	French	A1	0.301	0.4559	0.3974	0.2754
Baseline	French	1	0.2952	0.3760	0.3674	0.2626
RETUYT-INCO	French	A2	0.2764	0.4278	0.3776	0.2662
GMU	French	A2	0.0845	0.2394	0.1725	0.149
Archaeology	French	2	0.072	0.1704	0.1447	0.121
Archaeology	French	1	0.065	0.1265	0.1044	0.0819

Team Name	Language	ID	Acc@1@Top1	Acc@3@Top1	MAP@3	MAP@5
TMU-HIT	German	A1	0.4885	0.6695	0.4882	0.3548
TMU-HIT	German	A2	0.4411	0.6695	0.481	0.3504
GMU	German	A1	0.42	0.5817	0.4002	0.2874
GMU	German	1	0.4192	0.5824	0.4004	0.2874
TMU-HIT	German	A3	0.355	0.6695	0.4633	0.3398
RETUYT-INCO	German	A1	0.3022	0.434	0.2699	0.1787
RETUYT-INCO	German	A2	0.2671	0.4165	0.2626	0.1765
ISEP	German	1	0.2187	0.25	0.1984	0.1344
Baseline	German	1	0.1719	0.2192	0.1562	0.1054
GMU	German	2	0.1192	0.3	0.1852	0.1463
Archaeology	German	1	0.0614	0.114	0.0626	0.0484
Archaeology	German	2	0.028	0.0771	0.0388	0.0294

Team Name	Language	ID	Acc@1@Top1	Acc@3@Top1	MAP@3	MAP@5
TMU-HIT	Italian	A1	0.4762	0.7188	0.5661	0.4126
TMU-HIT	Italian	A2	0.4657	0.7188	0.558	0.4078
ISEP	Italian	1	0.4245	0.6614	0.5064	0.3788
GMU	Italian	A1	0.4042	0.6309	0.4615	0.3328
GMU	Italian	1	0.4035	0.6315	0.4616	0.3328
TMU-HIT	Italian	A3	0.3708	0.7188	0.5454	0.4002
RETUYT-INCO	Italian	A1	0.3163	0.4973	0.3511	0.2434
RETUYT-INCO	Italian	A2	0.3022	0.485	0.3305	0.2253
Baseline	Italian	1	0.2964	0.4684	0.3310	0.2254
GMU	Italian	A2	0.1546	0.3567	0.246	0.1965
Archaeology	Italian	2	0.0947	0.1929	0.1145	0.092
Archaeology	Italian	1	0.0491	0.1508	0.0975	0.0755

Team Name	Language	ID	Acc@1@Top1	Acc@3@Top1	MAP@3	MAP@5
TMU-HIT	Japanese	1	0.4	0.5771	0.4883	0.3588
TMU-HIT	Japanese	A1	0.3989	0.5764	0.4881	0.3586
TMU-HIT	Japanese	2	0.3824	0.5771	0.4779	0.3526
GMU	Japanese	A1	0.2583	0.4393	0.3618	0.2599
GMU	Japanese	1	0.2578	0.4385	0.3612	0.2595
Baseline	Japanese	1	0.1561	0.2421	0.1735	0.1173
GMU	Japanese	A2	0.1195	0.2847	0.2144	0.171
RETUYT-INCO	Japanese	A1	0.0949	0.137	0.1026	0.0665
RETUYT-INCO	Japanese	A2	0.0878	0.1405	0.0949	0.0607
Archaeology	Japanese	2	0.0368	0.0929	0.0592	0.0441
Archaeology	Japanese	1	0.0263	0.0824	0.0516	0.0391

Team Name	Language	ID	Acc@1@Top1	Acc@3@Top1	MAP@3	MAP@5
ISEP	Portuguese	1	0.485	0.6684	0.3538	0.2421
TMU-HIT	Portuguese	A1	0.4432	0.6595	0.3451	0.2285
TMU-HIT	Portuguese	A2	0.4095	0.6595	0.3341	0.2219
TMU-HIT	Portuguese	A3	0.3776	0.6595	0.3297	0.2193
Baseline	Portuguese	1	0.3509	0.4973	0.2330	0.1516
RETUYT-INCO	Portuguese	2	0.2768	0.4514	0.2094	0.136
RETUYT-INCO	Portuguese	A1	0.2748	0.4503	0.2088	0.1356
RETUYT-INCO	Portuguese	A2	0.2606	0.4202	0.207	0.1341
Archaeology	Portuguese	2	0.097	0.2539	0.092	0.0704
Archaeology	Portuguese	1	0.0864	0.2116	0.079	0.0574

Team Name	**Language**	**ID**	**Acc@1@Top1**	**Acc@3@Top1**	**MAP@3**	**MAP@5**
GMU	Sinhala	A1	0.2284	0.3163	0.1387	0.0894
GMU	Sinhala	1	0.2283	0.32	0.14	0.0902
TMU-HIT	Sinhala	A2	0.2214	0.3585	0.1673	0.108
TMU-HIT	Sinhala	A1	0.2144	0.3585	0.1709	0.1101
GMU	Sinhala	A2	0.13	0.3057	0.1147	0.0759
TMU-HIT	Sinhala	A3	0.1195	0.3585	0.1469	0.0957
Archaeology	Sinhala	1	0.0466	0.0783	0.0359	0.0242
ANU	Sinhala	1	0.0133	0.0166	0.0074	0.0045
RETUYT-INCO	Sinhala	A1	0.0017	0.0017	0.0041	0.0024
Archaeology	Sinhala	2	0	0	0	0
RETUYT-INCO	Sinhala	A2	0	0	0.0032	0.0019
Baseline	Sinhala	1	0.0000	0.0033	0.0028	0.0017

Team Name	**Language**	**ID**	**Acc@1@Top1**	**Acc@3@Top1**	**MAP@3**	**MAP@5**
TMU-HIT	Spanish	A1	0.4536	0.6526	0.6763	0.5276
TMU-HIT	Spanish	A2	0.4502	0.6526	0.6721	0.5251
GMU	Spanish	1	0.4182	0.6087	0.5987	0.4653
GMU	Spanish	A1	0.4165	0.6053	0.5948	0.4627
TMU-HIT	Spanish	A3	0.3642	0.6526	0.6592	0.5174
RETUYT-INCO	Spanish	3	0.3288	0.4839	0.4124	0.298
Baseline	Spanish	1	0.3254	0.4519	0.4157	0.3019
RETUYT-INCO	Spanish	A1	0.3187	0.4957	0.4075	0.2879
RETUYT-INCO	Spanish	A2	0.3069	0.4688	0.399	0.2789
GMU	Spanish	A2	0.236	0.4704	0.4371	0.3542
Archaeology	Spanish	2	0.0674	0.1736	0.1565	0.1292
Archaeology	Spanish	1	0.0455	0.1112	0.0951	0.0756

TMU-HIT at MLSP 2024:
How Well Can GPT-4 Tackle Multilingual Lexical Simplification?

Taisei Enomoto[†*], **Hwichan Kim**[†*], **Tosho Hirasawa**[†], **Yoshinari Nagai**[†]
Ayako Sato[†], **Kyotaro Nakajima**[†], **Mamoru Komachi**[‡]
[†]Tokyo Metropolitan University, ‡Hitotsubashi University
{enomoto-taisei@ed., kim-hwichan@ed., toshosan@, nagai-yoshinari@ed.,
sato-ayako@ed., nakajima-kyotaro@ed.}tmu.ac.jp, mamoru.komachi@hit-u.ac.jp

Abstract

Lexical simplification (LS) is a process of replacing complex words with simpler alternatives to help readers understand sentences seamlessly. This process is divided into two primary subtasks: assessing word complexities and replacing high-complexity words with simpler alternatives. Employing task-specific supervised data to train models is a prevalent strategy for addressing these subtasks. However, such approach cannot be employed for low-resource languages. Therefore, this paper introduces a multilingual LS pipeline system that does not rely on supervised data. Specifically, we have developed systems based on GPT-4 for each subtask. Our systems demonstrated top-class performance on both tasks in many languages. The results indicate that GPT-4 can effectively assess lexical complexity and simplify complex words in a multilingual context with high quality. The code used in our experiments is available at the following URL [1].

1 Introduction

The presence of unfamiliar words within a sentence can significantly impede its comprehension for readers. Such complex words may cause misunderstandings of the sentence's content or result in wasted time as readers may find themselves compelled to consult definitions of unfamiliar words. The development of a system capable of automatically simplifying complex words would enable readers to proceed without interruption. To achieve this, it is essential to first identify complex words and then replace them with more comprehensible alternatives. Numerous researchers have been undertaken focusing on each challenge, engaging in specialized endeavors known as Lexical Complexity Prediction (LCP) (Paetzold and Specia, 2016; Shardlow et al., 2021) and Lexical Simplification (LS) (McCarthy and Navigli, 2007; Specia et al., 2012; Saggion et al., 2022).

LCP is a task that assesses the complexity of a target word, i.e. its level of difficulty for understanding. Various methodologies have been proposed to tackle this task. A classical strategy is the frequency-based approach (Kajiwara and Komachi, 2018), which attributes higher complexity scores to words of lower frequency. Given the availability of supervised data, one viable option is to train a regression model to evaluate the word's complexity (Bani Yaseen et al., 2021; Pan et al., 2021). However, such abundant linguistic resources for supervised learning are scarce for many languages (Joshi et al., 2020). Therefore, there exists a need for an approach capable of determining lexical complexity without reliance on supervised data.

LS is a task that replaces a complex word with easier synonyms while maintaining the original meaning or information of the sentence. The LS pipeline comprises three primary components (North et al., 2023): substitute generation (SG), substitute selection (SS), and substitute ranking (SR). SG returns several candidate alternative words for the target word in a given sentence. SS then shifts through these alternatives, eliminating those unsuitable for the target word, such as more complex words or words with different parts of speech. SR sorts the candidate alternatives to prioritize words more suitable as alternatives, ensuring they appear higher in the ranking. Recent LS studies (Qiang et al., 2019; Przybyła and Shardlow, 2020) have often employed pre-trained language models, and systems using GPT-3.5 (Aumiller and Gertz, 2022) have demonstrated particularly effective performance.

This paper outlines our approach to the MLSP-2024 Shared Task (Shardlow et al., 2024a). Our system was based on GPT-4 and exclusively utilized the trial data provided by the organizers (Bott et al., 2024; North et al., 2024; Shardlow et al.,

[*]These two authors contributed equally to this work.
[1]https://github.com/tmu-nlp/GPT4MLSP

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 590–598
June 20, 2024 ©2024 Association for Computational Linguistics

2024b), without incorporating any supplementary datasets. In the LCP task, our system achieved the highest Pearson's correlations in nine out of the ten languages. Additionally, in the LS task, our system exhibited the highest performance in seven out of the ten languages. The findings indicate that GPT-4 has capabilities of assessing lexical complexity and replacing complex words with simpler alternatives in a multilingual context with a high level of quality.

2 LCP Subtask

In Lexical Complexity Prediction (LCP), we assessed the complexities using GPT-4[2] based system. Specifically, we employed G-EVAL (Liu et al., 2023), a scoring framework using large language models with a chain-of-thoughts, for our system.

2.1 G-EVAL for LCP

We constructed a base prompt following G-EVAL. Specifically, we first manually designed a prompt that contains the definition of the LCP task and the desired evaluation criteria as follows:

> *You will be given a sentence and a word included in the sentence. Your task is to rate the word on one metric. Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.*
>
> *Evaluation Criteria:*
>
> *Complexity (0.0, 0.25, 0.5, 0.75, 1.0): the complexity of a word in terms of how difficult the word is to understand.*

Then, we added a line of "*Evaluation Steps:*" to the prompt and let GPT-4 generate the following chain-of-thoughts (CoT) automatically:

> *Evaluation steps:*
>
> *1. Read the sentence and word carefully to understand the context.*
>
> *2. Determine the complexity of the word based on the following criteria:*
>
> *- 0.0: The word is simple and easily understandable to most people.*
>
> *- 0.25: The word may have some complexity or be specific to a certain field, but can still be understood with some effort.*
>
> *- 0.5: The word is moderately complex and may require some background knowledge or explanation to understand fully.*
>
> *- 0.75: The word is quite complex and may be difficult to understand without significant knowledge or explanation.*
>
> *- 1.0: The word is extremely complex and likely only understood by experts or individuals with specialized knowledge.*
>
> *3. Assign a complexity rating to the word.*

We denote this prompt as P_{base}. We added a test example (sentence and target word) to P_{base} and let GPT-4 generate the complexities for the example n times. We used the average of those as the final complexity.

We had multiple options regarding the type of language to use for a prompt. Although the language of the test example is expected to be the most intuitive and effective, previous studies (Lin et al., 2022; Ahuja et al., 2023) demonstrated that English prompt achieves the best performance for most test languages. Furthermore, we manually and automatically translated P_{base} to Japanese and French, respectively, and compared performances of P_{base} and the translated prompt in each language using trial data. The Pearson's correlation of P_{base} and the translated prompt were 0.821 and 0.600 in Japanese 0.416 and 0.205 in French, respectively. Therefore, we used P_{base} regardless of languages.

2.2 Prompts to Specify Language and Role

In addition to P_{base}, we defined and added a prompt to specify the language of the test example. Specifically, we added "*Please assign a complexity rating to the* `LANG_NAME` *word*" to the end of P_{base} where `LANG_NAME` is a language name of a test example, such as *English*, *Japanese*, and *French*. We denote the prompt with the language as P_{lang}.

In our preliminary observation, the complexities generated by P_{base} distributed nearly 0.0 to 0.1, which means that almost all words are easy to understand for GPT-4. Furthermore, this distribution differed from that of the gold complexities as shown in Figure 1. One of the potential reasons is that GPT-4 is familiar with the target words unlike human annotators because it was pre-trained by massively data. To fill the gap between GPT-4

[2] We used `gpt-4-0613` following Liu et al. (2023) for LCP.

	Ca	En	Fil	Fr	De	It	Ja	Pr	Si	Es
P_{base}	**0.646**	0.733	0.462	0.416	**0.793**	0.615	**0.821**	0.836	**0.347**	0.641
P_{lang}	0.493	0.734	0.516	0.516	0.783	0.666	0.674	0.802	-0.077	**0.659**
P_{role}	0.470	**0.783**	0.513	0.513	0.740	0.537	0.794	**0.849**	0.292	0.654
$P_{\text{lang+role}}$	0.484	0.729	**0.595**	**0.595**	0.771	**0.672**	0.598	0.803	0.056	0.631

Table 1: Pearson correlations on trial datasets for each language. The best scores are indicated in bold.

	Pearson	Spearman	MAE	MSE	R2
Zero-shot (Run 1)	0.5609	0.5697	0.1771	0.0487	-0.3111
Three-shot (Run 2)	**0.6241**	**0.6215**	**0.1327**	**0.0280**	**0.2456**

Table 2: LCP results on the all language's test dataset. MAE and MSE denote Mean Absolute Error and Mean Squared Error.

and human annotators, we gave the role to GPT-4. Specifically, we added "*You are an individual without specialized knowledge or expertise in a specific area.*" to the first of P_{base}. We denote the prompt with the role as P_{role}.

We compared performances of P_{base}, P_{lang}, P_{role}, and $P_{\text{lang+role}}$, the prompt to which both of the language and role are added, per each language using trial data. Table 1 shows Pearson's correlations of each prompt per each language. The table indicates that the best prompts differ for each language.

2.3 Experiments

Experimental settings. We used the test datasets provided by Shardlow et al. (2024a)[3] for our evaluations. The datasets encompass those for ten languages, and a composite test dataset that amalgamates the individual datasets for all languages. For details about the languages and the size of each dataset, please refer to the Appendix.

For evaluation metrics, we employed both Pearson's and Spearman's correlations, Mean Absolute Error (MAE), Mean Squared Error (MSE), and R2 following Shardlow et al. (2021). We reported the performance of the composite test dataset.

We chose prompts for each language that achieved the highest Pearson's correlation in Table 1. We scored the complexities in zero- and three-shot settings.[4] In the three-shot setting, we randomly sampled three examples from the trial data.

Experimental results. Table 2 shows the result on the test set of all languages and indicates that the three-shot settings consistently outperform the zero-shot one. The findings indicate the importance of providing demonstration examples in LCP and suggest the possibility that performance will be enhanced by increasing the number of shots.

3 LS Subtask

In TSAR-2022 Shared Task (Saggion et al., 2022) of LS, the system using GPT-3.5 (Aumiller and Gertz, 2022) demonstrated a significant lead over other neural approaches such as those using mask language models. Following these findings, we employed a GPT-based method using the latest available GPT-4[5] for LS.

3.1 Substitution Generation

The Base system. We manually designed a prompt [6] that instructs GPT-4 to generate ten alternative words for the target word as follows:

> *I will give you a* `LANG_NAME` *sentence and a word in the 'Sentence' and 'Word' format. List ten alternatives for the Word that are easier to understand, separated by ','.*
> *You must follow these four rules.*
> *1. Take into account the meaning of the Word in the Sentence.*
> *2. Alternatives must be easier to understand than the Word.*
> *3. Each alternative consists of one word.*
> *4. Do not generate an explanation.*

[3] `https://github.com/MLSP2024/MLSP_Data/tree/main`

[4] We indicated the hyperparameters, such as n, temperature, and frequency_penalty, in Table 4.

[5] We used `gpt-4-0125-preview` in LS experiments.

[6] We designed a specific prompt for the Japanese. Please refer to Appendix A for details.

	ACC@k@Top1			Potential@k				MAP@k		
	k=1	k=2	k=3	k=1	k=3	k=5	k=10	k=3	k=5	k=10
Base system (Run 1)	**0.3772**	**0.4919**	**0.5498**	**0.6739**	**0.8071**	**0.8407**	**0.8759**	**0.4652**	**0.3421**	**0.2026**
w/ Ranking$_{\mathrm{GPT-4}}$ (Run 2)	0.3573	0.4792	**0.5498**	0.6391	**0.8071**	**0.8407**	**0.8759**	0.4570	0.3371	0.2001
w/ Ranking$_{\mathrm{XGLM}}$ (Run 3)	0.2933	0.4554	**0.5498**	0.5918	**0.8071**	**0.8407**	**0.8759**	0.4461	0.3306	0.1969

Table 3: LS results on the test dataset for all languages.

The rules 3 and 4 are to ensure generating an alternative word consisting of a single word. We observed that GPT-4 generates "descriptions" rather than truly synonymous expressions without the rules. For instance, "neither positive nor negative" was generated as an alternative word for "neutrally." Since these "descriptions" were not appropriate as alternative words, we added the rules 3 and 4 to the prompt.

We let GPT-4 generate alternatives using the prompts for n times. Then, we ensemble the $n \times 10$ alternatives following Aumiller and Gertz (2022). We refer to this approach as "Base" (Run 1).

3.2 Substitution Ranking

We observed that the Base system exhibited high Potential@3 scores in the trial dataset [7], indicating that in numerous instances, at least one of the top three alternatives predicted by the system was present in the gold annotations. Therefore, we hypothesized that scores on metrics such as ACC@1 can be enhanced by re-ranking the top three words. In Run2 and Run3, we undertook the re-ranking of the top three alternatives for each instance from the Base system.

GPT-4-based re-ranking. Previous studies ranked alternative words based on their semantic similarity to the target word (Seneviratne et al., 2022; Whistely et al., 2022) or their familiarity to people (frequency of occurrence in a corpus) (Li et al., 2022; North et al., 2022). Following the studies, we designed two distinct prompts for re-ranking the generated alternatives in terms of semantic similarity to the target word and the alternatives' ease, respectively. We re-ranked the alternatives through each prompt and used a composite ranking as the final prediction. We refer to the approach as "Ranking$_{\mathrm{GPT-4}}$" (Run 2).

XGLM-based re-ranking. In addition, we hypothesized that words' preference varies between human annotators and GPT-4 due to disparities in the extent of knowledge accumulated. Therefore, we trained a re-ranking model to fill the gap and reflect annotators' preferences. Specifically, we performed an instruction-tuning of XGLM (Lin et al., 2022) using the trial data[8]. We re-ranked alternatives using the resulting model. We refer to this approach as "Ranking$_{\mathrm{XGLM}}$" (Run 3).

3.3 Experiments

Experimental settings. We employed the same datasets as described in Subsection 2.3 for evaluation. For evaluation metrics, we used ACC@k@Top1, Potential@k, and MAP@k following Saggion et al. (2022).

Experimental results and discussions. Table 3 shows results on the test set of all languages. The Base system outperformed the re-ranking systems, and this trend held in nine out of the ten languages except for Sinhala.

These results indicate that the ranking of alternatives generated by GPT-4 within the Base system is comparatively appropriate, whereas Ranking$_{\mathrm{GPT-4}}$ and Ranking$_{\mathrm{XGLM}}$ do not yield appropriate rankings. Notably, the scores of Ranking$_{\mathrm{XGLM}}$ are significantly degraded, suggesting that it is difficult to train a re-ranking model using only the trial data (i.e. 30 examples for each language). Developing a better re-ranking strategy is one of the challenges to further enhance the scores.

4 Conclusion

In this paper, we introduced GPT-4-based systems designed to assess word complexities and replace complex words with simpler ones. Our systems achieved superior performance in multiple languages for both LCP and LS tasks within MLSP-2024 Shared Task.

[7] Table 6 shows the scores of the Base system on the trial dataset for each language.

[8] The details about how to create instruction-tuning data are described in Appendix B.

To score complexities, we created a base prompt following G-EVAL (Liu et al., 2023) and added to the base prompt supplementary prompts to delineate the language of the test example and the role of the LLM. Our prompt, when applied within a three-shot setting, consistently achieved the highest Pearson's correlation across the majority of languages. Furthermore, our experiments suggest the potential for performance enhancement through the augmentation of few-shot examples. Therefore, we plan to explore the change in performance resulting from an increment in the number of few-shot examples.

For the task of replacing complex words with simpler alternatives, we manually crafted prompts. The experimental results indicate that these prompts yield alternatives of commendable quality. Additionally, we explored the possibility of enhancing the selection of generated alternatives by employing a re-ranking strategy using either GPT-4 or XGLM that were instruction-tuned by trial data. However, the re-raking approaches degraded the scores compared to the ones before re-ranking. For future work, we plan to devise an improved re-ranking methodology.

5 Limitations

Our approach leverages the OpenAI API, which can be costly. In order to make Lexical Simplification easily available to many users, it might be essential to devise an approach built on open-source models, achieves comparable performance to this study.

Acknowledgments

This work was supported by JST, PRESTO Grant Number JPMJPR2366, Japan, and the National Institute of Information and Communications Technology (NICT) under the Research and Development of externally controllable modeling of multimodal information to enhance the accuracy of automatic translation.

References

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, Kalika Bali, and Sunayana Sitaram. 2023. MEGA: Multilingual evaluation of generative AI. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4232–4267, Singapore. Association for Computational Linguistics.

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.

Tuqa Bani Yaseen, Qusai Ismail, Sarah Al-Omari, Eslam Al-Sobh, and Malak Abdullah. 2021. JUST-BLUE at SemEval-2021 task 1: Predicting lexical complexity using BERT and RoBERTa pre-trained language models. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 661–666, Online. Association for Computational Linguistics.

Stefan Bott, Horacio Saggion, Nelson Peréz Rojas, Martin Solis Salazar, and Saul Calderon Ramirez. 2024. Multils-sp/ca: Lexical complexity prediction and lexical simplification resources for catalan and spanish. *Preprint*, arXiv:2404.07814.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Tomoyuki Kajiwara and Mamoru Komachi. 2018. Complex word identification based on frequency in a learner corpus. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 195–199, New Orleans, Louisiana. Association for Computational Linguistics.

Xiaofei Li, Daniel Wiechmann, Yu Qiao, and Elma Kerz. 2022. MANTIS at TSAR-2022 shared task: Improved unsupervised lexical simplification with pretrained encoders. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 243–250, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval:

NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Diana McCarthy and Roberto Navigli. 2007. SemEval-2007 task 10: English lexical substitution task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 48–53, Prague, Czech Republic. Association for Computational Linguistics.

Kai North, Alphaeus Dmonte, Tharindu Ranasinghe, and Marcos Zampieri. 2022. GMU-WLV at TSAR-2022 shared task: Evaluating lexical simplification models. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 264–270, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Kai North, Tharindu Ranasinghe, Matthew Shardlow, and Marcos Zampieri. 2023. Deep learning approaches to lexical simplification: A survey. *Preprint*, arXiv:2305.12000.

Kai North, Tharindu Ranasinghe, Matthew Shardlow, and Marcos Zampieri. 2024. Multils: A multi-task lexical simplification framework. *arXiv preprint arXiv:2402.14972*.

Gustavo Paetzold and Lucia Specia. 2016. SemEval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569, San Diego, California. Association for Computational Linguistics.

Chunguang Pan, Bingyan Song, Shengguang Wang, and Zhipeng Luo. 2021. DeepBlueAI at SemEval-2021 task 1: Lexical complexity prediction with a deep ensemble approach. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 578–584, Online. Association for Computational Linguistics.

Piotr Przybyła and Matthew Shardlow. 2020. Multi-word lexical simplification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1435–1446, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, and Xindong Wu. 2019. Lexical simplification with pretrained encoders. In *AAAI Conference on Artificial Intelligence*.

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.

Sandaru Seneviratne, Elena Daskalaki, and Hanna Suominen. 2022. CILS at TSAR-2022 shared task: Investigating the applicability of lexical substitution methods for lexical simplification. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 207–212, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Huelsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Peréz Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, Sanja Štajner, Marcos Zampieri, and Horacio Saggion. 2024a. The BEA 2024 Shared Task on the Multilingual Lexical Simplification Pipeline. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*.

Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Huelsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Peréz Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, Marcos Zampieri, and Horacio Saggion. 2024b. An Extensible Massively Multilingual Lexical Simplification Pipeline Dataset using the MultiLS Framework. In *Proceedings of the 3rd Workshop on Tools and Resources for People with REAding DIfficulties (READI)*.

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.

Lucia Specia, Sujay Kumar Jauhar, and Rada Mihalcea. 2012. SemEval-2012 task 1: English lexical simplification. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 347–355, Montréal, Canada. Association for Computational Linguistics.

Peniel 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 the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 213–217, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

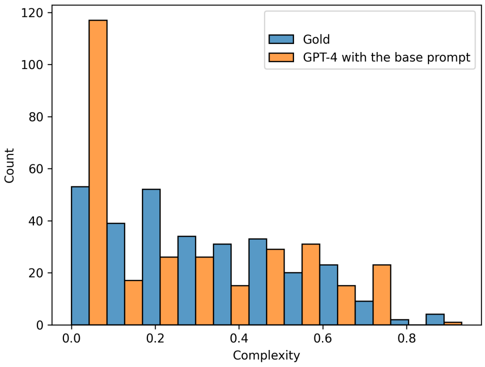


Figure 1: The histograms of the gold complexities and those derived from GPT-4 using the base prompt P_{base}. This figure shows that the complexities generated by GPT-4 are distributed predominantly within the range of 0.0 to 0.1.

	LCP	LS		
		SG	SG (ja)	SR
temperature	0.7	0.7	0.7	0.7
frequency_penalty	0.0	0.5	0.0	0.0
presence_penalty	0.0	0.3	0.0	0.0
n	20	10	10	10

Table 4: Hyperparameters

A Japanese Specific Prompt

In the case of Japanese, instead of generating only alternative words, we instructed GPT-4 to generate sentences in which the target word was replaced with each alternative word. Unlike the other nine languages, Japanese doesn't have spaces between words. Additionally, Japanese verbs, adjectives, adjectival verbs and auxiliary verbs undergo "Katsuyou" (inflection), wherein the ending of a word changes depending on the subsequent word. Some target words in the Japanese dataset are in Katsuyou-form; for instance, "募集し" is in the Katsuyou-form, while "募集する" is in the Basic-form. We observed that when we instructed GPT-4 to generate alternative words for a target word in Katsuyou-form, it often generated words in Katsuyou-form that did not suit the sentence or words in Basic-form. On the other hand, when we instructed GPT-4 to generate sentences in which the target word was replaced with each alternative word, GPT-4 could generate alternative words that have the correct Katsuyou-form to fit the sentence. Table 7 shows examples of GPT-4 outputs for each method. The details of the prompt are shown in Table 8.

Language	Number of Examples
English	570
Catalan	445
French	570
German	570
Spanish	593
Italian	570
Portuguese	569
Filipino	570
Japanese	570
Sinhala	600

Table 5: The size of test datasets.

Language	ACC@1	Potential@k		
		k=3	k=5	k=10
Catalan	0.600	0.866	0.866	0.900
English	0.766	0.833	0.866	0.866
Filipino	0.566	0.633	0.633	0.700
French	0.866	0.966	0.966	0.966
German	0.800	0.933	0.933	0.933
Italian	0.866	0.933	0.933	0.933
Japanese	0.800	0.966	0.966	0.966
Portuguese	0.666	0.766	0.800	0.900
Sinhala	0.600	0.733	0.766	0.800
Spanish	0.766	0.833	0.866	0.900

Table 6: LS results on the trial dataset for each language.

B Dataset Creation for Instruction-Tuning of XGLM

The alternative words listed as gold are ranked by frequency of suggestion by the annotators. We used this ranking to create data for instruction-tuning of XGLM from the trial data in eight languages except Filipino and Sinhala. [9] The query of the created data consisted of a contextual sentence, a target word, two alternative words in the trial data, and an instruction letting a model select a more suitable alternative word. The answer was the alternative with the highest ranking among the two alternatives. When Alternative 1 was ranked higher than Alternative 2 in the trial data, the template is as follows:

> *### Instruction* : *I will give you a* {LANG_NAME} *sentence, a word contained in the sentence and alternatives*

[9] Since Filipino and Sinhala are not included in the XGLM pre-training data, we exclude these languages from the fine-tuning data.

Sentence	ドラマに関する感想を募集し、週ごとにピックアップして回答も掲載した。
Target word	募集し
Gold	集め, 促し, 募り, 探し, 集めて, 呼びかけ, 広く集め, 呼びかけて, たくさん求め, 書いてもらい, ...
Only word	集めています, 求めています, 探しています, 募っています, 応募を受け付けています, 呼びかけています, 求めている, 探している, 求人しています, 集めている
With sent	集め, 求め, 探し, 求めて, 探して, 招待し, 募って, 要求し, 呼びかけ, 呼びかけて

Table 7: Examples of GPT-4 output in Japanese. “Gold” represents the correct answer in the trial data. “Only word” and “With sent” represent outputs when we instructed GPT-4 to generate ten alternative words and sentences where the target words are replaced with each alternative word, respectively.

> *for the word in the ‘Sentence’, ‘Word’ and ‘Alternatives’ format. Choose a more suitable alternative word to the Word in the Sentence.*
> *### Sentence*： `{SENTENCE}`
> *### Word*： `{TARGET_WORD}`
> *### Alternatives*： `{ALTERNATIVE 1, ALTERNATIVE 2}`
> *### Response*： `{ALTERNATIVE 1}`

We conducted re-ranking by employing a XGLM instruction-tuned on this dataset to predict the portion following *“### Response:”*.

Setting	Prompt Template
SG (non ja)	*I will give you a* {LANG_NAME} *sentence and a word in the 'Sentence' and 'Word' format. List ten alternatives for the Word that are easier to understand, separated by ','.* *You must follow these four rules.* *1. Take into account the meaning of the Word in the Sentence.* *2. Alternatives must be easier to understand than the Word.* *3. Each alternative consists of one word.* *4. Do not generate an explanation.* *Sentence:* {SENTENCE} *Word:* {TARGET_WORD} *Alternatives:*
SG (ja)	*I will give you a Japanese sentence and a word in the 'Sentence' and 'Word' format. Think ten easier alternatives for the Word in the Sentence. Then, output sentences where you have replaced the Word with each alternative enclosed by '**'.* *You must follow these three rules.* *1. Take into account the meaning of the Word in the Sentence.* *2. Alternatives must be easier to understand than the Word.* *3. Do not generate an explanation.* *Sentence:* {SENTENCE} *Word:* {TARGET_WORD} *Alternative sentences:*

SR (ease)	*I will give you a* {LANG_NAME} *sentence, a word and alternatives for the word in the 'Sentence', 'Word' and 'Alternatives' format. Arrange the Alternatives in order of their ease. Do not generate an explanation.*
	Sentence: {SENTENCE}
	Word: {TARGET_WORD}
	Alternatives: {ALTERNATIVES}
	Sorted Alternatives:

SR (sim)	*I will give you a* {LANG_NAME} *sentence, a word and alternatives for the word in the 'Sentence', 'Word' and 'Alternatives' format. Arrange the Alternatives in order of their semantic similarity to the Word, taking into account the meaning of the Words in the Sentence. Do not generate an explanation.*
	Sentence: {SENTENCE}
	Word: {TARGET_WORD}
	Alternatives: {ALTERNATIVES}
	Sorted Alternatives:

Table 8: Prompt templates used for GPT-4 in LS experiments. "SG" and "SR" represent the Substitute Generation and Substitute Ranking, respectively. LANG_NAME is empty when the language is English. In SG (ja), SENTENCE is a sentence with the target word encloseed by '**'. In SR, "ease" represents ranking based on ease of each alternative word, and "sim" represents ranking based on semantic similarity of each alternative word to the target word.

ANU at MLSP-2024: Prompt-based Lexical Simplification for English and Sinhala

Sandaru Seneviratne[1] **and Hanna Suominen**[1,2]
[1]The Australian National University (ANU) / Canberra, ACT, Australia
[2]University of Turku / Turku, Finland
Firstname.Lastname@anu.edu.au

Abstract

Lexical simplification, the process of simplifying complex content in text without any modifications to the syntactical structure of text, plays a crucial role in enhancing comprehension and accessibility. This paper presents an approach to lexical simplification that relies on the capabilities of generative Artificial Intelligence (AI) models to predict the complexity of words and substitute complex words with simpler alternatives. Early lexical simplification methods predominantly relied on rule-based approaches, transitioning gradually to machine learning and deep learning techniques, leveraging contextual embeddings from large language models. However, the the emergence of generative AI models revolutionized the landscape of natural language processing, including lexical simplification. In this study, we proposed a straightforward yet effective method that employs generative AI models for both predicting lexical complexity and generating appropriate substitutions. To predict lexical complexity, we adopted three distinct types of prompt templates, while for lexical substitution, we employed three prompt templates alongside an ensemble approach. Extending our experimentation to include both English and Sinhala data, our approach demonstrated comparable performance across both languages, with particular strengths in lexical substitution.

1 Introduction

Lexical simplification, an essential component in making complex text more understandable, involves replacing complex words with simpler alternatives while preserving the meaning and syntax (Bott and Saggion, 2011; Seneviratne et al., 2022b). This task is specifically valuable for people with limited knowledge in certain languages or domains or for people with low literacy skills (Gooding and Kochmar, 2019). Lexical simplification can be composed as a cascade of complex word identification and lexical substitution. Addressing both these tasks is vital for improved language understandability.

Complex word identification task is the first step in lexical simplification (Shardlow, 2014). This task can be formulated as identifying the complex words in text or as predicting the level of lexical complexity for each word. Various techniques have been employed for this task, ranging from rule-based (Devlin, 1998; Carroll et al., 1999) through threshold-based (Zeng et al., 2005) to lexicon-based approaches (Miller, 1995). Following these methods, researchers have also explored feature-based machine learning methods (Wróbel, 2016; Malmasi et al., 2016) that also incorporate word embedding models and more sophisticated approaches like deep learning models such as long short-term memory (LSTM) networks, modelling the problem as a sequence labelling task (Gooding and Kochmar, 2019). Recently, contextual embedding models like Bidirectional Encoder Representation from Transformers (BERT) have gained attention for complex word identification due to their ability to capture nuanced contextual information (Qiang et al., 2021; Seneviratne, 2024).

Similar to complex word identification, lexical substitution is an important sub-task for lexical simplification. Early methods relied on lexical resources to generate simpler, suitable, relevant substitutes for complex or target words (Biran et al., 2011; Pavlick and Callison-Burch, 2016). This evolved with the introduction of word embedding models like Word2Vec (Mikolov et al., 2013), Global Vectors for Word Representation (GloVe) (Pennington et al., 2014), and Embedding from Language Models (ELMo) (Peters et al., 2018). More recently, contextual embedding models such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and XLNet (Yang et al., 2019) have become popular for lexical substitution, sometimes combined with lexical resources for improved performance (Seneviratne et al., 2022a).

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 599–604
June 20, 2024 ©2024 Association for Computational Linguistics

Even though many natural language processing tasks have relied on more complex or sophisticated methods based on deep learning models and contextual embeddings, with the emergence of generative Artificial Intelligence (AI), most of the methods have shifted to exploring simpler approaches based on prompt engineering (Aumiller and Gertz, 2022). Prompt engineering presents straightforward and effective approaches for a wide range of tasks, including for lexical complexity prediction and substitution. In this study, we leveraged prompt engineering for both tasks, focusing on improving the accuracy and efficiency.

2 Experiments

2.1 Datasets

Our experiments and evaluations used the English and Sinhala language datasets provided by the MLSP-2024 shared task (Shardlow et al., 2024b; North et al., 2024).

Lexical Complexity Prediction. The trial subset of both English and Sinhala lexical complexity prediction datasets comprised 30 sentences each, and consisted of samples with the context, target word, and their respective lexical complexities. The data from the trial subset of the data was used for one-shot and few-shot prompt template creation. The test subset of the dataset consisted of 600 samples each for both English and Sinhala, which was used for the evaluation of the proposed prompt-based methods.

For lexical complexity prediction, since the dataset had samples where the same sentence had been associated with different target words, we first grouped the sentences together and obtained lexical complexities for each target word in a sentence. This facilitated a comparative perspective on the complexity levels of the target words relative to one another. Moreover, this enhanced the information included in the prompt template allowing a better understanding of the distinctions and variations in lexical complexity.

Lexical Substitution. For the lexical substitution task, we employed datasets in both English and Sinhala, each consisting of context sentences with words requiring simplification, along with sets of alternative words. Similar to the complexity prediction task, the trial subset of the both the datasets consisted of 30 samples, which were used for prompt template creation. The test subset of the data, that was used for evaluation, comprised 570 samples for English and 600 samples for Sinhala, respectively.

2.2 Methods

We relied on prompt-based methods for both lexical complexity prediction and lexical simplification through substitution generation. We relied on Generative Pre-trained Transformer– GPT3.5-turbo-instruct model with a *temperature* of 0.5 and *top_p* value of 1 for our experiments. This specific model has a context window of $4,096$ tokens.

Lexical Complexity Prediction. For lexical complexity prediction, we explored the following three distinct prompt templates to study how varying levels of additional information can affect the final prediction: zero-shot, one-shot, and few-shot. Each of these widely recognized templates provided unique information as to how additional contextual information influences lexical complexity prediction. Namely, the zero-shot template, which only used the given sample input to determine lexical complexity of the target word, served us as a baseline to compare with the other two prompt-template methods. For the one-shot approach, we selected a single random sample from the processed trial dataset. Conversely, the few-shot approach involved incorporating three examples from the trial dataset into the prompt. Since we processed the dataset to consolidate the same contexts and their target words, the samples included in the prompt consisted of context sentence along with their target words and the lexical complexity values.

Context: {context}
Question: Given the above context, give the lexical complexity for each word as a value between 0 and 1. The words are {words}
Lexical complexity:

Table 1: Zero-shot prompt template used for lexical complexity prediction. For one-shot and few-shot prompt templates, examples were incorporated.

Lexical Substitution for Simplification. Similar to the lexical complexity prediction task, we relied on three prompt templates for the initial generation of simpler, relevant, and suitable substitutes for a given target work. While our zero-shot approach only included the given context and the target word for substitution generation, we incorporated in the one-shot and few-shot prompt templates one and three samples from the trial dataset, respectively. In the latter two approaches, our prompt included the given context sentence, target word, and their possible substitutes for the generation process. In each prompt template, we asked the model to provide ten simpler substitutes for the target word.

Context: {context}
Question: Given the above context, list ten alternative words for {word} that are easier to understand.
Alternative susbtitutes:

Table 2: Zero-shot prompt template used for lexical substitution. For one-shot and few-shot prompt templates, examples were incorporated.

We further filtered the results obtained from the three prompt templates. To combine the results from the prompt templates, we followed (Aumiller and Gertz, 2022), where the authors computed a combination score V (Eq. 1) for each distinct prediction, where $\text{rank}_{Sp(s)}$ is the ranked position of a possible substitute s for a given prompt p.

$$V(s) = \sum_{p=1}^{3} \max(5.5 - 0.5 \times \text{rank}_{Sp(s)}, 0). \quad (1)$$

2.3 Evaluation metrics

We based the evaluation of the proposed methods on the metrics used in the MLSP-2024 shared task (Shardlow et al., 2024a). For lexical complexity prediction, Pearson's R, Spearman's Rank, *Mean Absolute Error* (MAE), and *Mean Squared Error* (MSE) were used. For the lexical substitution task, we relied on Accuracy@K ($K \in \{1, 2, 3\}$), *Potential@K* ($K \in \{1, 3\}$), and *Mean Average Precision@K* (MAP@K) ($K \in \{3, 5\}$).

3 Results

The results of the prompt-based lexical complexity prediction methods did not reach the performance levels of the top submissions in the lexical complexity prediction task (Table 3). While the best submission achieved Person's R of 0.8497, the best system from our experiments — the zero-shot approach — had Person's R of 0.3358. Among our prompt-template-based methods for Sinhala, the one-shot approach yielded the most promising results. However, its Pearson's R of 0.0534 was placed fifth among the submissions for Sinhala.

In lexical simplification for English, our proposed few-shot approach showed strong performance, achieving comparable results with respect to the best submissions for the task across all metrics (Table 4). The proposed method gave the Accuracy@1 score of 0.5105, while the best submission gave 0.5245. However, for Sinhala, our submission (which was the ensemble approach) did not show satisfactory performance.

4 Discussion

In this paper, we have explored the applicability of prompt-templates for both lexical complexity prediction and lexical substitution for simplification in English and Sinhala. Our investigation primarily focused on three prompting methodologies: zero-shot, one-shot, and few-shot. The experiments demonstrated diverse performance levels across the two tasks and languages under consideration.

The most effective approach of our experiments for predicting complexity in English relied on the zero-shot method, while for Sinhala, the one-shot approach gave the best results. This difference may stem from differences in language data availability and the complexity of each language and task. Compared to Sinhala, English has more language data available, providing the model with a more extensive information base. This could be a reason why for English the zero-shot approach performed better, as the model could leverage enough contextual information. However, Sinhala, being less extensively studied, likely has fewer linguistic resources and data available for training. Therefore, the one-shot approach, which provides additional context, may be better suited to capture the patterns and dependencies in the language.

Considering the performance of the prompt-based methods for complexity prediction in Sinhala, the few-shot approach did not perform as well as the one-shot approach, even though more additional information was provided. This discrepancy could be attributed to the quality of the samples included in the prompt template. If the chosen examples fail to adequately represent the lexical features and patterns of the language, it may lead to a degradation in performance, resulting in poorer results compared to the one-shot approach.

The results from the lexical substitution for simplification indicated varied performance. In English, out of our experiments, the few-shot approach gave the best results, closely followed by the ensemble approach, which combined results from all three prompt templates. This suggests that the few-shot approach provided good example instances that helped in capturing the lexical intricacies of the language. Therefore, while the ensemble approach gave comparable performance,

Team Name	Run ID	Pearson's R	Spearman's Rank	Mean Absolute Error	Mean Squared Error
English					
GMU	1	0.8497	0.7984	0.1137	0.0175
TMU-HIT	2	0.8198	0.7552	0.1108	0.0178
SDJZUandUU	3	0.8123	0.7754	0.1071	0.0175
RETUYT-INCO	2	0.5502	0.4923	0.1561	0.0328
ANU	1	0.3358	0.3591	0.3484	0.1478
GMU	A	0.3118	0.3183	0.1389	0.0346
CocoNut	1	0.1972	0.2160	0.4150	0.2263
Sinhala					
TMU-HIT	A	0.3081	0.3343	0.1666	0.0422
TMU-HIT	A	0.2482	0.3261	0.2126	0.0661
RETUYT-INCO	A	0.1344	0.1094	0.3355	0.1340
GMU	1	0.1246	0.1303	0.1018	0.0168
ANU	2	0.0534	0.0866	0.1741	0.0539
SCaLAR	A	0.0450	0.0279	0.1576	0.0321
Archaeology	2	0.0437	0.0298	0.1239	0.0236
GMU	A	0.0263	0.0284	0.1066	0.0180

Table 3: Results of the experimented approaches on the test subsets of the English and Sinhala datasets provided at the MLSP-2024 shared task for lexical complexity prediction.

Team Name	Run ID	Accuracy@1	Accuracy@2	Accuracy@3	Potential@1	Potential@3	MAP@3	MAP@5
English								
TMU-HIT	1, A1	0.5245	0.6807	0.7456	0.7982	0.9035	0.5762	0.4142
GMU	1, A1	0.5157	0.635	0.6894	0.7491	0.8754	0.513	0.3691
ANU	3	0.5105	0.6175	0.6649	0.7684	0.8824	0.5324	0.3744
ANU	1	0.4684	0.5929	0.6561	0.735	0.8684	0.5069	0.3652
ISEP_Presidency	1	0.4684	0.607	0.6754	0.7649	0.8859	0.5351	0.3877
ANU	2	0.4631	0.5807	0.6421	0.7228	0.8614	0.4978	0.3524
TMU-HIT	2	0.4438	0.6298	0.7456	0.7333	0.9035	0.5595	0.4042
RETUYT-INCO	3	0.3789	0.5105	0.5701	0.5947	0.7824	0.3832	0.2634
RETUYT-INCO	2	0.3438	0.4701	0.5526	0.5789	0.7666	0.3718	0.2542
Sinhala								
GMU	A1	0.2284	0.2829	0.3163	0.311	0.4165	0.1387	0.0894
GMU	1	0.2283	0.2866	0.32	0.3116	0.4183	0.14	0.0902
TMU-HIT	A2	0.2214	0.3286	0.3585	0.3198	0.4903	0.1673	0.108
TMU-HIT	A1	0.2144	0.304	0.3585	0.3444	0.4903	0.1709	0.1101
GMU	A2	0.13	0.2372	0.3057	0.195	0.3848	0.1147	0.0759
TMU-HIT	A3	0.1195	0.2759	0.3585	0.2249	0.4903	0.1469	0.0957
Archaeology	1	0.0466	0.0633	0.0783	0.0666	0.1383	0.0359	0.0242
ANU	1	0.0133	0.015	0.0166	0.0133	0.0183	0.0074	0.0045
RETUYT-INCO	A1	0.0017	0.0017	0.0017	0.0123	0.0123	0.0041	0.0024
RETUYT-INCO	A2	0	0	0	0.0087	0.0105	0.0032	0.0019

Table 4: Results of the experimented approaches on the test subsets of the English and Sinhala datasets provided at the MLSP-2024 shared task for lexical susbtitution for simplification.

it did not filter the best predictions as effectively as the few-shot method. However, for Sinhala lexical substitution, we only employed the ensemble approach. Unfortunately, the results indicated subpar performance. This suggests that the ensemble approach did not effectively capture the lexical patterns, dependencies of Sinhala language, that resulted in unsatisfactory outcomes.

The findings indicate the importance of investigating the influence of the factors such as data availability, language complexity, and sample quality on the outcomes of lexical simplification tasks. Additionally, refining prompt tuning methods could enhance the effectiveness and outcomes.

5 Conclusion

In this work, we have used prompt-based methods for both lexical complexity prediction and lexical substitution for simplification, focusing on exploring the applicability of generative AI methods. The results from the different methods indicate varied performance levels across the two tasks and languages, giving evidence of challenges related to data availability, representations, quality of the samples, language complexity, and adaptability of the models for the lexical simplification task. This encourages further investigations that could potentially improve the performance differences.

6 Limitations

The experiments were conducted using GPT-based models, which posed challenges primarily due to their significant resource requirements (Aumiller and Gertz, 2022). Thus, to facilitate these experiments, we accessed the GPT model through an Application Programming Interface (API), which costed approximately $8 for all experiments. Furthermore, the utilization of these models raises ethical concerns surrounding data privacy and transparency limitations. Additionally, our findings highlighted variations in results based on the prompt template, the examples included in the prompts, and the parameters used, highlighting the need for further investigation on the usability of these models for NLP tasks.

Acknowledgments

This research was funded by and has been delivered in partnership with Our Health in Our Hands (OHIOH), a strategic initiative of the ANU, which aims to transform health care by developing new personalized health technologies and solutions in collaboration with patients, clinicians and healthcare providers.

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.

Or Biran, Samuel Brody, and Noémie Elhadad. 2011. Putting it simply: a context-aware approach to lexical simplification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 496–501, Portland, Oregon, USA. Association for Computational Linguistics.

Stefan Bott and Horacio Saggion. 2011. An unsupervised alignment algorithm for text simplification corpus construction. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 20–26, Portland, Oregon. Association for Computational Linguistics.

John Carroll, Guido Minnen, Darren Pearce, Yvonne Canning, Siobhan Devlin, and John Tait. 1999. Simplifying text for language-impaired readers. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 269–270, Bergen, Norway. 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.

Siobhan Devlin. 1998. The use of a psycholinguistic database in the simplification of text for aphasic readers. *Linguistic Databases*.

Sian Gooding and Ekaterina Kochmar. 2019. 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.

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*.

Shervin Malmasi, Mark Dras, and Marcos Zampieri. 2016. LTG at SemEval-2016 task 11: Complex word identification with classifier ensembles. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 996–1000, San Diego, California. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*.

George A Miller. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Kai North, Tharindu Ranasinghe, Matthew Shardlow, and Marcos Zampieri. 2024. Multils: A multi-task lexical simplification framework. *arXiv preprint arXiv:2402.14972*.

Ellie Pavlick and Chris Callison-Burch. 2016. Simple PPDB: A paraphrase database for simplification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 143–148, Berlin, Germany. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, Yang Shi, and Xindong Wu. 2021. LSBert: Lexical simplification based on BERT. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3064–3076.

Ishanee Sandaru Seneviratne. 2024. *Text Simplification Using Natural Language Processing and Machine Learning for Better Language Understandability*. Ph.D. thesis, The Australian National University.

Sandaru Seneviratne, Elena Daskalaki, Artem Lenskiy, and Hanna Suominen. 2022a. CILex: An investigation of context information for lexical substitution methods. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4124–4135, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Sandaru Seneviratne, Elena Daskalaki, and Hanna Suominen. 2022b. CILS at TSAR-2022 shared task: Investigating the applicability of lexical substitution methods for lexical simplification. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 207–212, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, 4(1):58–70.

Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Huelsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Peréz Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, Sanja Štajner, Marcos Zampieri, and Horacio Saggion. 2024a. The BEA 2024 Shared Task on the Multilingual Lexical Simplification Pipeline. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*.

Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Huelsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Peréz Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, Marcos Zampieri, and Horacio Saggion. 2024b. An Extensible Massively Multilingual Lexical Simplification Pipeline Dataset using the MultiLS Framework. In *Proceedings of the 3rd Workshop on Tools and Resources for People with REAding DIfficulties (READI)*.

Krzysztof Wróbel. 2016. PLUJAGH at SemEval-2016 task 11: Simple system for complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 953–957, San Diego, California. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Qing Zeng, Eunjung Kim, Jon Crowell, and Tony Tse. 2005. A text corpora-based estimation of the familiarity of health terminology. In *Biological and Medical Data Analysis: 6th International Symposium, ISBMDA 2005, Aveiro, Portugal, November 10-11, 2005. Proceedings 6*, pages 184–192. Springer.

ISEP_Presidency_University at MLSP 2024 Shared Task: Using GPT-3.5 to Generate Substitutes for Lexical Simplification

Benjamin Dutilleul[1], **Mathis Debaillon**[1], and **Sandeep Mathias**[2,3]

[1]Institut Supérieur d'électronique de Paris,
[2]Information Retrieval Lab, Department of Computer Science & Engineering
[3]Presidency University, Bangalore
Correspondence: sandeepalbert@presidencyuniversity.in

Abstract

Lexical substitute generation is a task where we generate substitutes for a given word to fit in the required context. It is one of the main steps for automatic lexical simplification. In this paper, we introduce an automatic lexical simplification system using the GPT-3 large language model. The system generates simplified candidate substitutions for complex words to aid readability and comprehension for the reader. The paper describes the system that we submitted for the Multilingual Lexical Simplification Pipeline Shared Task at the 2024 BEA Workshop. During the shared task, we experimented with Catalan, English, French, Italian, Portuguese, and German for the Lexical Simplification Shared Task. We achieved the best results in Catalan and Portuguese, and were runners-up in English, French and Italian. To further research in this domain, we also release our code upon acceptance of the paper[1].

1 Introduction

Test simplification is an important educational application. It aims to simplify text to make the generated simpler text easier for reading and comprehension by different readers who may be either young learners, people with language disabilities (Eg. aphasia), second-language learners, etc. A lot of the research done in the area of text simplification is split into mainly 2 parts, namely syntactic simplification and lexical simplification.

Syntactic simplification involves splitting the sentences into smaller sentences (Klerke et al., 2016). Lexical simplification, on the other hand, involves simplifying the text by replacing more complex words and phrases with simpler, and in context, synonyms (Shardlow, 2014).

The lexical simplification pipeline consists of multiple sub-tasks, (Shardlow, 2014) as shown in Figure 1. These subtasks are complex word identification (where we identify which word we have to consider for simplification), substitution generation (where we generate candidate synonyms for the given complex word), substitution selection (where we select the candidate synonyms which are contextually correct), and substitution ranking (where we rank the selected candidates from easiest to most complex).

With the advent of large language models (LLMs) like GPT-3, the potential for automating this task has increased significantly. These models, trained on vast amounts of text, have shown remarkable proficiency in understanding context and generating human-like text. Unlike pre-trained language models like BERT (Devlin et al., 2019), LLMs are significantly harder to fine-tune due to the massively larger number of parameters (BERT has about 110 million parameters, while GPT-3 has about 175 **billion** parameters). Because of this, we use GPT-3 using prompt-engineering, where we provide a prompt to the system to generate substitutes.

1.1 Organization of the Paper

The rest of the paper is organized as follows. We define the problem statement of our work in Section 2. Section 3 summarizes some of the recent related work in this domain. We discuss the different datasets used in Section 4. We describe our system in Section 5. Our results are reported and discussed in Section 6 and we conclude our paper and mention future work in Section 7.

2 Problem Statement

The Multilingual Lexical Simplification Pipeline (MLSP) Shared Task dealt with 2 problems. The first was Lexical Complexity Prediction (LCP). In this task, the participants had to develop a system where they were given a context and word in a

[1] The code for the paper is available at: `https://github.com/lwsam/ISEP-LS`

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 605–609
June 20, 2024 ©2024 Association for Computational Linguistics

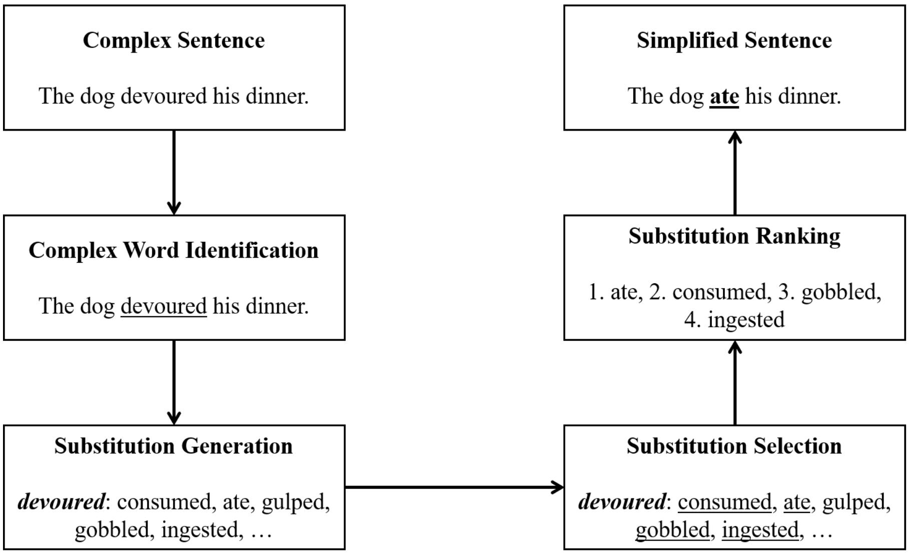

Complex Sentence
The dog devoured his dinner.
Complex Word Identification
The dog devoured his dinner.
Substitution Generation
devoured: consumed, ate, gulped, gobbled, ingested, …
Substitution Selection
devoured: consumed, ate, gulped, gobbled, ingested, …
Substitution Ranking
1. ate, 2. consumed, 3. gobbled, 4. ingested
Simplified Sentence
The dog ate his dinner.

Figure 1: Lexical Simplification Pipeline showing the different tasks traditionally used in lexical simplification. In this example, we simplify a complex sentence ("The dog devoured his dinner.") to a simplified sentence ("The dog **ate** his dinner.").

given language, and they had to assess how easy / complex the word was[2]. This is similar to the SemEval 2021 Shared Task on Lexical Complexity Prediction (Shardlow et al., 2021).

The second problem was called Lexical Simplification (LS), where we are given an input context and complex word and we need to generate a **ranked list** of upto 10 simplifications in increasing order of complexity (i.e. from the simplest substitute to the most complex substitute). This is similar to the 2022 Text Simplification, Accessibility and Readability Shared Task on Lexical Simplification (Saggion et al., 2022).

Both these problem statements required participants to build systems in multiple languages, such as English, Catalan, French, German, Italian, Brazilian Portuguese, Spanish, Bengali, Sinhala, Filipino, and Japanese. In our paper, we mainly focus on the second task (lexical simplification) for the first 6 languages listed (English to Brazilian Portuguese). More details on the shared task are available inn the shared task report (Shardlow et al., 2024).

3 Related Work

There has been a number of shared tasks dealing with different aspects of the lexical simplification pipeline.

For complex word identification, one of the earliest shared tasks was held in 2016 (Paetzold and Specia, 2016a). The winners of that shared task used a system of soft voting with different "voters", where the voters are either lexicon based, threshold-based, or machine-learning assisted (Paetzold and Specia, 2016b). In 2018, another shared task on complex word identification was held as part of the BEA Workshop collocated with NAACL (Yimam et al., 2018). It had a monolingual track and multilingual tracks where the systems would be tested on German, French and Spanish. The winning team (Gooding and Kochmar, 2018), used a similar approach as Paetzold and Specia (2016b), but with a much wider range of features.

One of the challenges is trying to assess a score for how simple / complex a word is, given the context. This step is critical for complex word identification. In light of this, Shardlow et al. (2021) conducted a lexical complexity prediction task at SemEval 2021.

The advent of LLMs inspired a significant change in task specification. The 2022 TSAR Shared Task on Lexical Simplification (Saggion et al., 2022) had 3 languages - English, Spanish and Brazilian Portuguese. The participants in that shared task had to generate a set of substitutes for each language. While some systems such as Whistely et al. (2022) used a procedure of candidate generation (using pre-trained language models like BERT (Devlin et al., 2019)), cosine similar-

[2] We attempted to participate in this task as well, but due to some issues with the formatting in the output, we were unable to make a good submission by the shared task deadline.

ity and part-of-speech tagging as filters, the winning team (Aumiller and Gertz, 2022) used prompt-engineering on a large language model.

4 Datasets

Language	Test Set Size
English	570
Catalan	445
French	570
German	570
Italian	570
Portuguese	569

Table 1: Sizes of the testing dataset for each language.

For the shared task, participants were provided only with **trial** data. That is, a very few context-complex word pairs. Each language had a trial dataset of 30 context-complex word pairs[3]. Our systems were then evaluated on test sets of varying sizes. Table 1 shows the sizes of each language's testing dataset.

5 Experiment

5.1 System Used

For our system, we utilize Open AI's GPT 3.5 model[4]. We use a maximum of 256 tokens in the prompt with a frequency penalty of 0.5 and a presence penalty of 0.3.

The first step that we do is detect the language of the context. Based on the language chosen, we select a prompt for simplification. If no language is detected, then we default to the English prompt.

Once we detect the language, we next generate the prompt from a set of templates. We use 3 types of templates, similar to (Aumiller and Gertz, 2022).

5.2 Types of Prompts

Context-Free Prompt. This is a prompt that asks for synonyms of the complex word without providing any context. This tests the model's general knowledge of synonyms generation.

Context-Free Prompt. Template: "Give me ten simplified synonyms for the following word {complex word}". **Example:** "Give me ten simplified synonyms for the following word {**distraught**}"

Zero-Shot Prompt. This type of prompt provides the context and the complex word, and asks the LLM for simpler synonyms without any additional examples. This is used to gauge the model's ability to generate synonyms based solely on the given context and complex word.

Zero-shot Prompt. Template: "Context: {context} Question: Given the above context, list ten alternative words for {complex word} that are easier to understand. Answer:" **Example:** "Context: {*After Ron nearly dies drinking poisoned mead that was apparently intended for Professor Dumbledore, Hermione becomes so distraught that they end their feud for good.*} Question: Given the above context, list ten alternative words for {**distraught**} that are easier to understand. Answer:"

Single-shot Prompt. This is a prompt that includes one example of a complex word and its synonyms, followed by the target complex word. This aims to guide the model by showing an example of the desired output.

Single-shot Prompt. Template: "Question: Find ten easier words for **prerequisite**. Answer: 1. requirement 2. required 3. essential 4. need 5. precondition 6. prior 7. necessary 8. necessity 9. prior 10. prescribed. Question: Find ten easier words for {**complex word**}. Answer:"

Few-Shot Prompt. This is similar to the single-shot prompt, but with multiple examples provided to give the model a clearer understanding of the task.

5.3 Prompting the LLM

For each generated prompt, we send a request to the GPT-3.5 API. The predictions from GPT-3.5 are cleaned. Predictions from different prompts are aggregated and ranked and the top (at most) 10 synonyms are submitted as the output for our system.

5.4 Evaluation Metrics

We used the same evaluation metrics as given in the shared task. However, in Section 6, we report an **aggregate** of the evaluation metrics.

The different evaluation metrics used for automatic evaluation are:

- **MAP@K**. This metric uses an ordered list of gold-standard substitutes to compare the system output with. This metric takes into

[3] NOTE: The number of contexts for most of the languages are less than 30, as some contexts were repeated with different complex words.

[4] Model name - gpt-3.5-turbo-instruct-0914

English		Catalan		French	
System	Performance	System	Performance	System	
TMU-HIT	0.677	ISEP_PU	0.547	TMU-HIT	0.697
ISEP_PU	0.643	TMU-HIT	0.524	ISEP_PU	0.660
GMU	0.639	GMU	0.445	GMU	0.590
ANU	0.636	RETUYT-INCO	0.397	RETUYT-INCO	0.497
RETUYT-INCO	0.530	Archaeology	0.215	Archaeology	0.258
CocoNut	0.386	—	—	—	—
Archaeology	0.288	—	—	—	—
German		**Italian**		**Portuguese**	
System	**Performance**	**System**	**Performance**	**System**	**Performance**
TMU-HIT	0.626	TMU-HIT	0.673	ISEP_PU	0.571
GMU	0.548	ISEP_PU	0.635	TMU-HIT	0.551
RETUYT-INCO	0.413	GMU	0.607	RETUYT-INCO	0.379
ISEP_PU	0.257	RETUYT-INCO	0.225	Archaeology	0.230
Archaeology	0.142	Archaeology	0.225	—	—

Table 2: Results of our system compared with the best performances from all other systems based on the **mean** of all the evaluation metrics. Our system is highlighted in blue. Due to space constraints, we refer to it as "ISEP_PU".

account the ranking of each of the generated outputs. Here, $K = \{3, 5, 10\}$.

- **Accuracy@k@top1**. This is the percentage of instances, where, out of the top k outputs given by the system, at least one of them matches the top gold-standard substitute. Here, $k = \{1, 2, 3\}$.

- **Potential@k**. This is similar to the MAP@K metric, where we take $k = \{3, 5, 10\}$.

Based on the above metrics, we calculate our aggregate metric, **Performance**, which is the **arithmetic mean** of the other metrics.

6 Results and Analysis

We report the results of our experiments in Table 2. From the above table, we observe that we perform quite well compared to other systems, in almost all the languages except for German. We have achieved the best performances in Catalan and Brazilian Portuguese, as well as the second-best performances in English, French and Italian.

One of the challenges that we faced was in constructing the prompts for different languages. While the authors of the paper are L1 / fluent speakers of English and French, we needed the help of Google Translate to translate the prompts from English to other languages like German / Italian / Portuguese.

One of the challenges of using LLMs currently is that they are computationally intensive, requiring hundreds of GB of GPU power to fine-tune. Another challenge is that the current LLMs are focused on generating ranked substitutes irrespective of the target user. For example, young learners may have different requirements for simplification, as opposed to second-language learners, or people with reading disabilities. This can be tackled by modifying the prompts (especially the one-shot / few-shot prompts) to generate different simplifications based on the target user.

7 Conclusion and Future Work

Although we have performed reasonably well in the shared task for lexical simplification, we would like to extend our work for other languages which we were not able to participate in. Most of the other languages possess orthographic challenges because they do not use the Roman script, such as Bengali, Japanese, etc.

In the future, we would also like to focus on instruction tuning to improve the performance for personalizing the LLM for simplification. Currently, the predictions from the LLM are independent of the user. This means that a system built using this approach may generate the same output irresppective of the user the text should be simplified for. One method for resolving this is to utilize a user's cognitive information to try to perform complex word identification, as well as generate and rank candidate simplifications.

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.

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. 2018. CAMB at CWI shared task 2018: Complex word identification with ensemble-based voting. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 184–194, New Orleans, Louisiana. Association for Computational Linguistics.

Sigrid Klerke, Yoav Goldberg, and Anders Søgaard. 2016. Improving sentence compression by learning to predict gaze. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1528–1533, San Diego, California. Association for Computational Linguistics.

Gustavo Paetzold and Lucia Specia. 2016a. SemEval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569, San Diego, California. Association for Computational Linguistics.

Gustavo Paetzold and Lucia Specia. 2016b. SV000gg at SemEval-2016 task 11: Heavy gauge complex word identification with system voting. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 969–974, San Diego, California. Association for Computational Linguistics.

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.

Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*.

Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Huelsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Peréz Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, Sanja Štajner, Marcos Zampieri, and Horacio Saggion. 2024. The BEA 2024 Shared Task on the Multilingual Lexical Simplification Pipeline. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*.

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.

Peniel 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 the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 213–217, Abu Dhabi, United Arab Emirates (Virtual). 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.

Archaeology at MLSP 2024: Machine Translation for Lexical Complexity Prediction and Lexical Simplification

Petru Theodor Cristea
petru-theodor.cristea@s.unibuc.ro

Sergiu Nisioi
sergiu.nisioi@unibuc.ro

Human Language Technologies Research Center
Faculty of Mathematics and Computer Science
University of Bucharest

Abstract

We present the submissions of team Archaeology for the Lexical Simplification and Lexical Complexity Prediction Shared Tasks at BEA2024. Our approach for this shared task consists in creating two pipelines for generating lexical substitutions and estimating the complexity: one using machine translated texts into English and one using the original language. For the LCP subtask, our xgb regressor is trained with engineered features (based primarily on English language resources) and shallow word structure features. For the LS subtask we use a locally-executed quantized LLM to generate candidates and sort them by complexity score computed using the pipeline designed for LCP. These pipelines provide distinct perspectives on the lexical simplification process, offering insights into the efficacy and limitations of employing Machine Translation versus direct processing on the original language data.

Our results and experiments are released at https://github.com/senisioi/MLSP_Participants

1 Introduction

In the realm of Natural Language Processing (NLP), the twin challenges of lexical complexity prediction and language simplification play pivotal roles in advancing text comprehension and promoting accessibility. Lexical complexity prediction refers to the difficulty of understanding phrases based on their lexical features, while simplification aims to enhance accessibility by offering simplified, easier-to-understand alternatives. The importance of addressing these challenges is underscored by their wide-ranging implications across various domains (Gooding, 2022; North et al., 2023; Saggion et al., 2023).

Our approach is guided by the idea to extend such methods beyond the languages that currently have available data sets or corpora; thus, our first set of submissions to the 2024 MLSP Shared Task (Shardlow et al., 2024a) uses machine translation to translate all datasets and languages into English, which has been the central language of text simplification and complexity research in recent years (North et al., 2023). Both the lexical simplification (LS) and the lexical complexity prediction (LCP) pipelines are using only data in English in this case [1].

The second approach is trained on the original texts as released by Shardlow et al. (2024b) and uses an LCP pipeline trained with language-independent hand-crafted features such as word length, syllables, vowels, etc. and a regression method trained on the small trial data from the original language.

For generating candidates for lexical simplification, we have opted for an LLM that can be run locally using a quantized version of OpenHermes 2.5 based on Mistral (Jiang et al., 2023) that has been fine-tuned on code. According to the authors[2], the model was trained on a good ratio of code instruction (7-14% of the total dataset) that boosted several noncode benchmarks, including TruthfulQA, AGIEval, and GPT4All suite. The quantized LLM is not inherently multilingual, however, in our small-scale tests we have observed some ability to generate simplification candidates for non-English language,

The LLM we used to generate the alternatives does not guarantee the correct form of the generated alternative and this problem is amplified by using Machine Translation to get the phrases in original languages, which could incorrectly translate words without context. Regarding Machine

[1] Because of a bug in our submission code, the first LCP submission was run with a LCP regression model trained purely on English data with no other language involved.

[2] https://huggingface.co/teknium/OpenHermes-2.5-Mistral-7B

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 610–617
June 20, 2024 ©2024 Association for Computational Linguistics

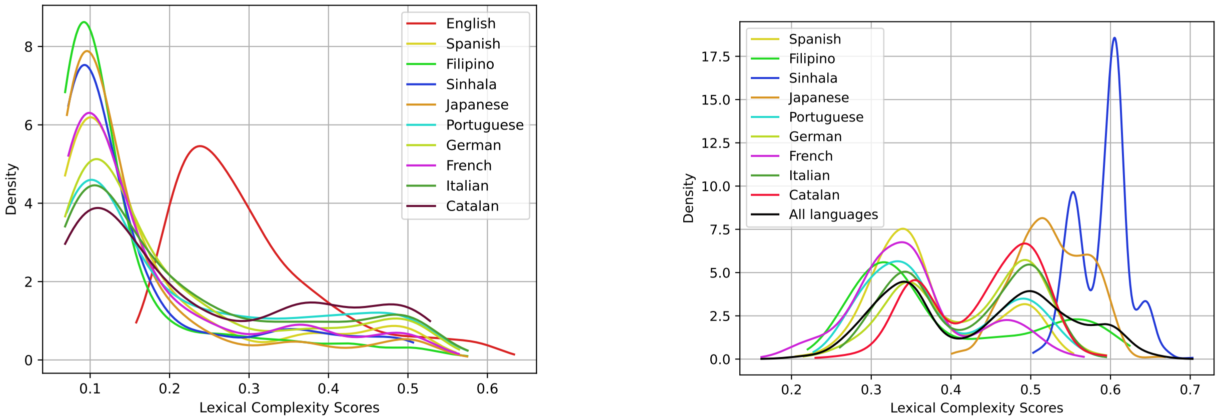


Figure 1: Density plots of LCP submitted scores predicted on the test set using (a) translated sentences and Englsh original sentences and (b) original-language texts. The model run on original texts generally observes two peaks, one with smaller values for simple words and one with larger values associated with complex words. Back-translated words show quite a different pattern with only a single peak of words marked as simple.

		es	fil	si	ja	pt	de	fr	it	ca	en
	mean	1.36	1.52	1.49	1.74	1.32	1.23	1.39	1.29	1.52	-
Trans.	**max**	8.2	6	12.71	8.75	8.25	5	6.5	4.3	6.3	-
	empty	0.2	0	1.3	0.4	0	0.4	0.5	0.5	0.7	-
	mean	1.27	1.33	1.14	2.3	1.22	1.11	1.29	1.19	1.39	1.31
Orig.	**max**	3.44	4.1	3	8.4	5.1	3.2	5.5	6.2	6	3.4
	empty	14.2	2.8	3.2	0.7	8.6	3.2	3.7	12.5	2	5.3

Table 1: Average lengths of multi-word expressions that our systems suggested as alternative lexical simplifications. Row empty indicate the percentage of empty suggestions for each language. The upper part of the table shows that the number of empty suggesions of OpenHermes2.5 are low for texts translated into English, but the average number of new words is higher than for prompts using texts in the original language.

Translation, we used DeepL[3] for French, Spanish, Japanese, German, Portuguese, Italian, and Google Translate for Sinhala, Catalan, and Filipino, thus obtaining only sentences in English to be able to effectively apply feature extraction.

In many cases, during the translation process, contextual information or expressions may be lost, significantly affecting the correlation between features. Table 1 shows the average number of multi-word expressions introduced by the translation step or by the predictions of the LLM model. Our LLM suggested in many cases empty strings, we did not check for those cases. As it stands, 14% of Spanish 12% of Italian are empty, however the overall scores with LLMs for these languages exceed the scores with MT (by a small margin). With MT, the number of empty suggestions is considerably smaller, but strangely enough 5% of original English sugestions are empty.

The LLM works better at generating candidates directly using English translations as the number of empty candidates is lower; however, the actual candidates generated tend to be multi-word expressions instead of simple lexical substitutions.

In summary, combining our approach for predicting lexical complexity and simplification in a unified framework may not be the best solution for text comprehension, but it can provide a source of interesting results for different languages.

2 Lexical Complexity Prediction

For lexical complexity prediction we reuse an approach that has been previously tested at the LCP2021 Shared Task (Shardlow et al., 2021) that obtained a Pearson correlation of .75 using a regression method trained on hand-crafted features.

Shallow Word Structure Features

We believe that this set of characteristics is as much as possible language independent when additional Latin-alphabet transliterations are used:

- character length of word

[3] https://deepl.com

- zipf_frequency from wordfreq library (Speer, 2022) (except for Sinhala)
- is title (not applicable for non-Latin glyphs)
- number of vowels (not applicable for non-Latin glyphs)
- number of syllables from pyphen library[4] (not applicable for non-Latin glyphs)

Medical Research Council Psycholinguistic Database

The MRC database (Wilson, 1988) is one of the most widely used feature source for LCP (Devlin, 1998; Yimam et al., 2018; Shardlow et al., 2021; North et al., 2023) demonstrating over three decades of high usability (Scott et al., 2019) built on top of word annotations (Thorndike and Lorge, 1944) and highlighting the necessity of such databses beyond the English language. Each lexical item is lemmatized using the spacy English large model (Montani et al., 2023) and searched in the database. The features we employ are:

- aoa - age of acquisition 1-7 Likert scale multiplied by 100 (Carroll and White, 1973; Gilhooly and Logie, 1980)
- conc - concreteness rating from the methodology of Spreen and Schulz (1966); Gilhooly and Logie (1980): "words referring to objects, materials, or persons were to receive a high concreteness rating, and words referring to abstract concepts that could not be experienced by the senses were to receive a low concreteness rating"
- fam - (Noble, 1953; Gilhooly and Logie, 1980) familiarity rating (100-700)
- imag - imagability / imagery rating (Paivio et al., 1968; Gilhooly and Logie, 1980): "words arousing images most readily were to be rated 7, and words arousing images with great difficulty or not at all were to be rated 1" scores multiplied by 100
- meanp - meaningfulness - defined as "the mean number of associations given in a 30-sec production period" from the Paivio et al. (1968)
- meanc - meaningfulness - Colorado Norms (Toglia and Battig, 1978) obtained using a different methodology from meanp (Wilson, 1988)
- brown_freq - Brown verbal frequency (Brown, 1984)
- Kucera-Francis number of categories, samples and frequency (Kučera et al., 1967)
- tl_freq - Thorndike-Lorge written frequency (Thorndike and Lorge, 1944)

Syntactic Features

For all lanugages except Filipino and Sinhala, we load spacy medium-sized models (Montani et al., 2023) using the latest version available. The only syntactic features are the number of immediate children in syntactic dependency parse. We use spacy here to introduce additional boolean features such as: is entity, is sentence start, is sentence end. Such words could be markers of conceptual complexity (Stajner et al., 2020).

WordNet Features

Similar to the MRC features, these are only available for English. We access WordNet (Miller, 1994) from NLTK (Bird et al., 2009) to extract the number of synsets, hypernyms, and hyponyms.

External Lists

The system also incorporates external datasets, such as the Dale-Chall (Dale and Chall, 1948) list to create a boolean feature set. Furthermore, additional frequency data is derived from non-native speakers in the European Parliament (Nisioi et al., 2016).

Similar features to ours have been used for the CWI identification Shared Task in 2018 (Gooding and Kochmar, 2018) obtaining excellent results on a related task.

Regression Model

We use an XGBoost Regressor (Chen and Guestrin, 2016), which operates within a gradient boosting framework, sequentially training weak learners to minimize a specified loss function. For this task, we do not employ hyperparameter tuning. All features are passed through a scikit learn standard scaler (Pedregosa et al., 2011) which standardizes the features to zero mean and a standard deviation of one. Although it might have been advisable to check which features are good for scaling, we did not proceed with this step, but rather passed all the features (including the Boolean ones) through the scaler.

We train our model on the English dataset released during LCP2021 (Shardlow et al., 2021) concatenated with all the languages from the current year's shared task (Shardlow et al., 2024b).

[4] https://doc.courtbouillon.org/pyphen

We use the same amount of features for all languages, which presents an interesting corner case where words with similar forms are found in the English-only resources. Such examples can be in Filipino: *amin* (En. *us*), *ate* (En.: *sister*) or the French words: *notice, question, coach*, Portuguese words such as: *bases, rigor*, and Catalan: *decimals*. This idea might point to a future research direction to explore where false friends, borrowings, and cognates (Dinu et al., 2023) could have the ability to preserve lexical traits across languages that have a history of contact.

3 Lexical simplification

For lexical simplification, we employ the locally run quantized OpenHermes 2.5 based on Mistral (Jiang et al., 2023) using llama-cpp[5] and langchain (Chase, 2022) libraries. The context contains the entire sentence and the target word, and the model is prompted to generate a json with potential replacement candidates. We run the model on the English-translated texts and the results are then back-translated into the initial language. The model prompt is as follows: *This sentence "TRANSLATION" is a translation of "ORIGINAL" and the word "TRANSLATED_WORD" is a translation of "ORIGINAL_WORD" Provide a list of 10 alternative simpler words (as a json object) that a child would understand easily to replace the word ""TRANSLATED_WORD"" in the following sentence. It is mandatory for pattern of the answer to be displayed as a JSON with words as keys and complexity scores as values with all the 10 alternatives.*

The second set of submissions is generated with the model running on the original language date. Nevertheless, it is imperative to acknowledge that the model's capability is constrained when handling multilingual data, often leading to hallucinations. The prompt used for original language data is: *Provide a list of 10 alternative simpler words (as a json object) that a child would understand easily to replace the word "ORIGINAL_WORD" in the context of the following sentence. It is mandatory to use suitable meanings for the context of the sentence and for the pattern of the answer to be displayed as a JSON with words as keys and complexity scores as values with all the 10 alternatives. Provide only words in "LANGUAGE". Sentence: "ORIGINAL. Here are some possible synonyms: "SYNONYMS"* The synonyms are given in the context extracted from ConceptNet (Speer et al., 2017) with a quick request to the API.

4 Results

Our first set of submissions (suffixed with "*_1.tsv") contain LCP only run with English-only models and LS predictions run on translated texts. The translation model tends to increase the number of words, as seen in Table 1 because we translate words out of their context, and some translations might not end up being found in the text mot-à-mot. We identify a target word in the context of the sentence (which will become our target for LCP) by doing a proximal cosine similarity search using spacy embeddings.

Our second set of submissions (suffixed with "*_2.tsv") are LCP predictions run on the original target words. Figure 1 shows the density plots of the predictions on the test set. The translations-to-English complexity scores (a) are in the same range for all langauges (except for Sinhala) while the predictions on the original texts (b) show more divergent patterns due to different features available for each language. Here we only report the results on the LCP task as these are the only ones that proved to be competitive in the shared task. For a complete set of results we point the reader to the official task page[6].

We perform several experiments on the trial data to verify which features of the original language have the strongest correlation with the complexity scores provided. This should give us a rough idea of the features that contribute the most to the final prediction. The correlations computed on the trial data are reported as a proxy for potential feature impact; given the small sample size, they may also show accidentally high values.

For **English**-language predictions (originals and translations included) word frequency achieves between -.64 and -.7 ρ, followed by MRC features, such as the Kucera-Francis (Kučera et al., 1967) number of categories feature (-.55 ρ). MRC features are generally well correlated among each other. Sparse features such as Dale-Chall, EuroParl frequency, hyponyms, synsets, and other MRC-based features contribute significantly because they create a boundary between words that are not in the external resource (feature value 0) and words

[5] https://github.com/ggerganov/llama.cpp

[6] https://sites.google.com/view/mlsp-sharedtask-2024/home

that are (value $>$ 0). For **French**: word frequency (-.4 ρ) and the number of immediate children in syntactic dependency parse (.4 ρ) show the best correlation with the complexity annotations. **German** trial data shows that word frequency is at -.76 ρ followed by character length .46 ρ, this could be a lucky coincidence from the small size or the distribution in the trial data. Similarly, **Filipino** (-.61 ρ), **Spanish** (-.6 ρ), **Portuguese** (-.71 ρ), and Italian (-.63 ρ) show relatively good correlations between complexity scores and word frequency. **Catalan** shows weak correlations of all individual characteristics (-.2 ρ on frequency), which also confirms our overall scores (Shardlow et al., 2024a), and so is **Japanese** (-.58 ρ). **Sinhala** is a special case of language where we do not use word frequency nor other resource and the only relevant features is the character length (-.3 ρ).

5 Conclusions

Translating documents into English and making lexical simplification predictions using translated texts introduces noise and severely limits the ability of the model to produce coherent substitutions, especially since our approaches with translations did not take into consideration the proper morphological form of the substitution or of the original word. Our results (reported in the Appendix) show that complexity prediction is significantly affected by the translation as much as lexical simplification. Our MT approach was surpassed by models trained only on English data, which appear to have a better ability to generate good LCP predictions on other languages (especially Latin-script languages or languages with a historical contact). This approach can yield decent results, achieving similar correlations to models trained directly on the source language or models using LLMs and transformer features (Shardlow et al., 2024a). Last but not least, we conclude that frequency- and string-based approaches might be a powerful alternative for LCP on low-resource languages.

6 Limitations

We observe several limitations of our approaches:

- potential innacuracies stemming from the translation system

- using MT for the low resource setting could be detrimental to the development of resources in the original language; translating all languages into English is not always feasible and depends on cultural factors, availability of resources and so on

- the performance of the MT systems themselves can vary depending on factors such as language pair, domain specificity, and the quality of the training data; in our case we have used closed-source models which is not desirable for open research

- our work is focused only on a single LLM that is English-centric, however the model was not able to generate suggestions that are in the correct tense or syntactic agreement with the rest of the sentence

7 Acknowledgements

This work has been supported by POCIDIF project in Action 1.2 "Romanian Hub for Artificial Intelligence".

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

Gordon DA Brown. 1984. A frequency count of 190,000 words in the london-lund corpus of english conversation. *Behavior Research Methods, Instruments, & Computers*, 16(6):502–532.

John B Carroll and Margaret N White. 1973. Age-of-acquisition norms for 220 picturable nouns. *Journal of Verbal Learning and Verbal Behavior*, 12(5):563–576.

Harrison Chase. 2022. LangChain. Software. Released on 2022-10-17.

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*, pages 785–794.

Edgar Dale and Jeanne S Chall. 1948. A formula for predicting readability: Instructions. *Educational research bulletin*, pages 37–54.

Siobhan Devlin. 1998. The use of a psycholinguistic database in the simplification of text for aphasic readers. *Linguistic databases*.

Liviu Dinu, Ana Uban, Alina Cristea, Anca Dinu, Ioan-Bogdan Iordache, Simona Georgescu, and Laurentiu Zoicas. 2023. RoBoCoP: A comprehensive ROmance BOrrowing COgnate package and benchmark for multilingual cognate identification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7610–7629, Singapore. Association for Computational Linguistics.

Ken J Gilhooly and Robert H Logie. 1980. Age-of-acquisition, imagery, concreteness, familiarity, and ambiguity measures for 1,944 words. *Behavior research methods & instrumentation*, 12(4):395–427.

Sian Gooding. 2022. On the ethical considerations of text simplification. *arXiv preprint arXiv:2204.09565*.

Sian Gooding and Ekaterina Kochmar. 2018. Camb at cwi shared task 2018: Complex word identification with ensemble-based voting. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 184–194.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *ArXiv*, abs/2310.06825.

Henry Kučera, Winthrop Francis, William Freeman Twaddell, Mary Lois Marckworth, Laura M Bell, and John Bissell Carroll. 1967. Computational analysis of present-day american english. *(No Title)*.

George A. Miller. 1994. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Ines Montani, Matthew Honnibal, Matthew Honnibal, Adriane Boyd, Sofie Van Landeghem, and Henning Peters. 2023. explosion/spaCy: v3.7.2: Fixes for APIs and requirements.

Sergiu Nisioi, Ella Rabinovich, Liviu P Dinu, and Shuly Wintner. 2016. A corpus of native, non-native and translated texts. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4197–4201.

Clyde E Noble. 1953. The meaning-familiarity relationship. *Psychological Review*, 60(2):89.

Kai North, Marcos Zampieri, and Matthew Shardlow. 2023. Lexical complexity prediction: An overview. *ACM Computing Surveys*, 55(9):1–42.

Allan Paivio, John C Yuille, and Stephen A Madigan. 1968. Concreteness, imagery, and meaningfulness values for 925 nouns. *Journal of experimental psychology*, 76(1p2):1.

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.

Horacio Saggion, Sanja Štajner, Daniel Ferrés, Kim Cheng Sheang, Matthew Shardlow, Kai North, and Marcos Zampieri. 2023. Findings of the tsar-2022 shared task on multilingual lexical simplification. *arXiv preprint arXiv:2302.02888*.

Graham G Scott, Anne Keitel, Marc Becirspahic, Bo Yao, and Sara C Sereno. 2019. The glasgow norms: Ratings of 5,500 words on nine scales. *Behavior research methods*, 51:1258–1270.

Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Huelsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Peréz Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, Sanja Štajner, Marcos Zampieri, and Horacio Saggion. 2024a. The BEA 2024 Shared Task on the Multilingual Lexical Simplification Pipeline. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*.

Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Huelsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Peréz Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, Marcos Zampieri, and Horacio Saggion. 2024b. An Extensible Massively Multilingual Lexical Simplification Pipeline Dataset using the MultiLS Framework. In *Proceedings of the 3rd Workshop on Tools and Resources for People with REAding DIfficulties (READI)*.

Matthew Shardlow, Richard Evans, Gustavo Henrique Paetzold, and Marcos Zampieri. 2021. Semeval-2021 task 1: Lexical complexity prediction. *arXiv preprint arXiv:2106.00473*.

Robyn Speer. 2022. rspeer/wordfreq: v3.0.

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.

Otfried Spreen and Rudolph W Schulz. 1966. Parameters of abstraction, meaningfulness, and pronunciability for 329 nouns. *Journal of Verbal Learning and Verbal Behavior*, 5(5):459–468.

Sanja Stajner, Sergiu Nisioi, and Ioana Hulpuș. 2020. CoCo: A tool for automatically assessing conceptual complexity of texts. In *Proceedings of the*

Twelfth Language Resources and Evaluation Conference, pages 7179–7186, Marseille, France. European Language Resources Association.

Edward Lee Thorndike and Irving Lorge. 1944. The teacher's word book of 30,000 words.

Michael P Toglia and William F Battig. 1978. *Handbook of semantic word norms.* Lawrence Erlbaum.

Michael Wilson. 1988. Mrc psycholinguistic database: Machine-usable dictionary, version 2.00. *Behavior research methods, instruments, & computers*, 20(1):6–10.

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

Language	Run ID	Pearson's R	Spearman's Rank	Mean Absolute Error	Mean Squared Error	R2
Catalan	1	0.2960	0.3029	0.1270	0.0246	-0.0342
Catalan	2	0.2744	0.2649	0.1236	0.0235	0.0110
Catalan	T	0.243333	0.200048	0.186026	0.050979	-1.146786
English	2	0.7904	0.7547	0.1225	0.0206	0.4393
Filipino	1	0.3620	0.4133	0.1729	0.0416	-0.9131
Filipino	2	0.4427	0.4476	0.1251	0.0234	-0.0763
Filipino	T	0.170322	0.200824	0.152792	0.039501	-0.817912
French	1	0.5335	0.5310	0.1898	0.0487	0.2136
French	2	0.4411	0.4188	0.1851	0.0504	0.1862
French	T	0.507726	0.502782	0.178938	0.046882	0.243141
German	1	0.5508	0.5726	0.1217	0.0252	0.0686
German	2	0.5577	0.5774	0.1369	0.0306	-0.1320
German	T	0.158362	0.18251	0.313923	0.129138	-3.779821
Italian	1	0.5341	0.5320	0.1705	0.0398	-0.4175
Italian	2	0.4790	0.4805	0.1426	0.0298	-0.0599
Italian	T	0.29937	0.309153	0.148348	0.03802	-0.353931
Japanese	1	0.2803	0.2648	0.2650	0.0894	-2.2358
Japanese	2	0.4851	0.5126	0.1440	0.0303	-0.0983
Japanese	T	0.038864	0.067513	0.181906	0.053068	-0.920658
Portuguese	1	0.7143	0.7102	0.1454	0.0276	-0.2612
Portuguese	2	0.6831	0.6923	0.1068	0.0166	0.2419
Portuguese	T	0.42688	0.446644	0.122814	0.026359	-0.206013
Sinhala	1	-0.0290	-0.0272	0.3920	0.1676	-9.3516
Sinhala	2	0.0437	0.0298	0.1239	0.0236	-0.4590
Sinhala	T	0.10023	0.065891	0.122526	0.028593	-0.76549
Spanish	1	0.5274	0.4793	0.1312	0.0265	0.2507
Spanish	2	0.5034	0.4588	0.1255	0.0272	0.2304
Spanish	T	0.326812	0.245494	0.20517	0.067601	-0.912674

Table 2: Lexical Complexity prediction of our models. The submissions marked with 1 are using a model trained only on the English language. The ones marked with 2 are trained on the entire multilingual data. And the ones marked with T are predictions only on translated data. It is clear from this table that translations significantly underperform predictions on original language even if the model was only trained on English data.

RETUYT-INCO at MLSP 2024: Experiments on Language Simplification using Embeddings, Classifiers and Large Language Models

Ignacio Sastre **Leandro Alfonso** **Facundo Fleitas**

Federico Gil **Andrés Lucas** **Tomás Spoturno**

Santiago Góngora **Aiala Rosá** **Luis Chiruzzo**

Instituto de Computación, Facultad de Ingeniería, Universidad de la República
Montevideo, Uruguay

Abstract

In this paper we present the participation of the RETUYT-INCO team at the BEA-MLSP 2024 shared task. We followed different approaches, from Multilayer Perceptron models with word embeddings to Large Language Models fine-tuned on different datasets: already existing, crowd-annotated, and synthetic. Our best models are based on fine-tuning Mistral-7B, either with a manually annotated dataset or with synthetic data.

1 Introduction

History has shown that technology can mean a step forward for inclusion and social development. For instance, NLP can change how different social groups interact with texts, by automatically adapting texts to the reader's needs and hence improving digital accessibility. One of the many NLP tasks devoted to this objective is *lexical simplification*, where systems are built to replace complex words by simpler ones. This has an immediate impact on language learners and children, but also on people with different types of learning or reading difficulties (Paetzold and Specia, 2016).

The BEA-MLSP 2024 shared task (Shardlow et al., 2024a) proposes an excellent opportunity to explore two problems related to this path: to score how complex a word is in a given context (task 1), and to find simpler substitutes for that word (task 2). The dataset used both as trial and test sets covers 10 different languages: Catalan, English, Filipino, French, German, Italian, Japanese, Portuguese, Sinhala and Spanish (Shardlow et al., 2024b). This dataset was annotated using the MultiLS Framework (North et al., 2024).

In this paper we present the participation of the Uruguayan RETUYT-INCO team at this shared task, describing the approaches followed and the datasets used. The main challenge to solve these tasks is the scarcity of data: only 30 examples for each language were given as trial data, and no training data. We decided to use the trial data as a development set to compare our experiments against each other, and rely on other sources of data (already existing datasets, crowd-sourced, or synthetic).

2 Related Work

Lexical complexity prediction and lexical simplification tasks have been addressed in different challenges in the past. We discuss the most recent ones for each task.

In the SemEval-2021 Task 1: Lexical Complexity Prediction (Shardlow et al., 2021), participants developed systems that, given a word within a sentence, assign it a complexity value on a continuous scale. An extended version of the CompLex Corpus (Shardlow et al., 2020) was used, with 10,800 instances of words and multi-word expressions scored according to their complexity. Deep Learning based systems performed the best, followed closely by feature-based approaches.

The TSAR-2022 Shared Task on Lexical Simplification (Saggion et al., 2022) hosted a shared task on Multilingual Lexical Simplification for English, Portuguese, and Spanish. The participants had to propose simpler substitutes for a complex word in a given context. Some trial examples were provided in each language (10 for English, 10 for Portuguese, and 12 for Spanish). The best results were obtained by approaches based on masked language models.

3 Approaches

In this section we detail the five different approaches followed. We experimented with static word embeddings, contextual embeddings, fine-tuning Mistral 7B on synthetic data, crowd-sourced data and existing data, and also with the Groq platform. We describe each of them next.

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 618–626
June 20, 2024 ©2024 Association for Computational Linguistics

3.1 Word Embeddings + Frequency Baselines

We created baseline approaches to the two tasks based on the use of word embeddings and word frequencies. In these baselines we prioritized using collections of embeddings and word frequency lists that were collected in the same way for all the languages in the task, so we used the Polyglot (Al-Rfou et al., 2013) word embedding collections, and the word frequency datasets collected from subtitles by Hermit Dave[1]. These resources are available for many languages, including all the languages in the shared task with the exception of the specific Filipino variety of the Tagalog language. In that case we used the corresponding resources for Tagalog, even if they could have some differences.

The approach for task 1 (Complex Word Prediction) is non-contextual, as no information from the context sentence is used: we take the 10 closest words to the target word in the embeddings collection, then use the frequency as a proxy to how complex a word is, assuming that more frequent words are simpler than less frequent ones. We sort the 10 closest words plus the target word by frequency and estimate the complexity of the target word as the relative position in this list, being 0 if it is the most frequent of the set and 1 if it is the least frequent.

The approach for task 2 (Lexical Simplification) was similar: finding the 10 most similar words to the target in the embeddings set, and sorting them by frequency. Besides the Polyglot embeddings and subtitle word frequency lists, for task 2 we also tried variants of this baseline approach using bigger and richer word embedding collections and frequency lists. For Spanish we used the `SBW-vectors-300-min5` embeddings[2] trained with the Spanish Billion Word Corpus[3]; for English the `googlenewsvectors` collection[4], and for Portuguese a word2vec collection trained from the ConLL17 corpus[5].

We also used other word frequency lists: for Spanish we used the Wiktionary Spanish frequency list[6], for English the Kaggle English Word Frequency dataset[7] compiled from the Google Web Trillion Word Corpus, and for Portuguese the frequency counts of the `wordfreq` library[8]. Another variant of this approach was sorting the replacement candidates by the distance with respect to the target word, without using word frequencies at all.

Besides these static word embedding approaches, we also tried with pre-trained contextual word embeddings such as BERT (Devlin et al., 2019) and ALBERT (Lan et al., 2019). We encode the context sentence and substitute the target word with the `[MASK]` token to obtain the 10 most probable replacements, that could be sorted either by probability or with word frequencies. In this case we used the BETO (Cañete et al., 2020) models `dccuchile/bert-base-spanish-wwm-cased` and `dccuchile/albert-xxlarge-spanish` for Spanish and HuggingFace models `google-bert/bert-large-cased` and `albert/albert-xlarge-v2` for English.

3.2 Fine-tuning Mistral 7B

This section presents two different approaches to fine-tuning an LLM to solve these tasks.

3.2.1 Fine-tuning on a Synthetic Dataset from Claude 3

It is well known that larger and more complex LLMs like the GPT family or Claude 3 Opus LLM from Anthropic[9] generally have good results in many NLP tasks. However, these are closed models, and we wanted to try if it was possible to at least distill some of their capabilities into a smaller model that is more resource-efficient, open and accessible to run in our available environment. To achieve this, and to alleviate the data scarcity problem, given that preliminary experiments with Claude 3 using the trial data showed promising results in a zero-shot scenario, we built a synthetic dataset using this LLM.

Generation of the synthetic data

Figure 1 shows a diagram of the synthetic dataset generation process. The complete prompts for each step can be found in Appendix C, while a comprehensive explanation of the entire process is provided in Appendix A. Below is a concise overview.

[1] https://github.com/hermitdave/FrequencyWords/

[2] https://github.com/dccuchile/spanish-word-embeddings

[3] https://crscardellino.github.io/SBWCE/

[4] https://www.kaggle.com/datasets/adarshsng/googlenewsvectors

[5] http://vectors.nlpl.eu/repository/

[6] https://en.wiktionary.org/wiki/User:\Matthias_Buchmeier#Spanish_frequency_list

[7] https://www.kaggle.com/datasets/rtatman/\english-word-frequency

[8] https://pypi.org/project/wordfreq/

[9] https://www.anthropic.com/news/claude-3-family

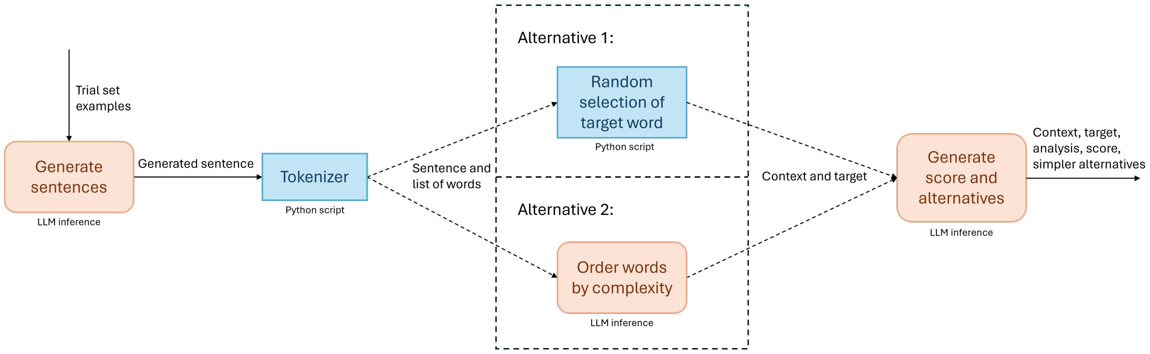


Figure 1: Diagram illustrating the process used to generate the synthetic dataset.

The process starts prompting the Claude model multiple times, including four random sentences from the trial dataset (i.e. following a few-shot strategy (Brown et al., 2020)) to get 250-500 sentences for each language. Then, each generated sentence is processed to generate different (context,target) pairs, as needed for the task 2 of the shared-task. Finally we generate the complexity score (1 to 5) and simpler alternatives for a given (context, target) pair, as needed for the task 1. In this final step we prompt the Claude model with an example of a word with a score of 1 and another one with a score of 5. Additionally we also include a Chain-of-Thought analysis (CoT, Wei et al. (2022)) to improve the performance of the model.

Each row of the resulting dataset consists of the context sentence, the target word, the analysis (CoT), the complexity score and the simpler alternatives. We elaborated a dataset of 2211 examples: 961 in Spanish, 750 in English and 500 in Portuguese. Our decision to focus on these three languages was due to time constraints and also because these are languages that we are familiar with, so we were able to check the overall quality of the synthetic text.

Fine-tuning details In order to fine-tune a smaller model for both task 1 and 2, each example of the dataset is transformed into a string which is a concatenation of the context sentence, the target word, the complexity score, and the simpler alternatives. Each of the parts is separated using XML tags, as can be observed in appendix C.4.

We tried adding the analysis (CoT) before the score when using the Spanish dataset. Table 3 and 4 in appendix B show the results for all the combinations of these techniques (CoT and SC). As can be seen, using a variation of SC without CoT gave the best results. Because of this, we decided to use this method for the rest of the languages.

These formatted examples are utilized for fine-tuning `Mistral 7B Instruct v0.2`[10]. Due to resource constraints, the model is 4-bit quantized and is fine-tuned using the Low-Rank Adaptation (LoRA) method (Hu et al., 2022).

Three different models were trained this way: using only the Spanish portion of the dataset, using Spanish and English, and using the whole dataset (Spanish, English and Portuguese). As a consequence, these were also the languages we focused more on evaluating. We also tried the last model on the Catalan language, given the similarity with Spanish and for testing the generalization capabilities of the fine-tuned model.

When doing inference with these models, a variation of the Self-Consistency technique (SC, Wang et al. (2023)) was employed. For the Lexical Simplification task, inference is conducted 10 times per example with a temperature of 0.7, resulting in up to 30 simpler alternatives with repetitions. These are then counted and arranged in order or frequency, with the most frequent ones appearing first, and any repeated occurrences are eliminated.

For the Lexical Complexity Prediction task, the prompt is structured such that the immediate next token represents the score, so only one token is sampled.This process is performed concurrently 100 times (within one batch) with temperature set to 1. The average score is then computed and normalized to the 0-1 range.

3.2.2 Fine-tuning on an existing English Dataset

We also tried another approach to fine-tune Mistral by curating the LCP2021 (Shardlow et al., 2020) dataset for English[11], recommended by the organiz-

[10] `https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2`

[11] `https://github.com/MMU-TDMLab/CompLex`

Experiment	Lang.	Pearson	Spearman	MAE	MSE	RMSE	r2
LoRA Mistral-7B, LCP2021	en	0.8061	0.7596	0.1405	0.0252	0.1587	0.3154
MLP with RoBERTa embeddings	en	0.5502	0.4923	0.1561	0.0328	0.1811	0.1062
LoRA Mistral7B, SC en-es dataset	en	0.7599	0.7406	0.1867	0.0433	0.2081	-0.1796
MLP with BETO embeddings	es	0.3126	0.2369	0.1433	0.0349	0.1868	0.0131
LoRA Mistral7B, SC en-es-pt dataset	es	0.6641	0.6547	0.1311	0.0254	0.1594	0.2808
LoRA Mistral7B, SC en-es-pt dataset	pt	0.6772	0.7121	0.2067	0.0557	0.2360	-1.5487
LoRA Mistral7B, SC en-es-pt dataset	ca	0.3948	0.3862	0.199	0.0569	0.2385	-1.3972
LoRA Mistral7B, SC en-es-pt dataset	all	0.4858	0.4892	0.2089	0.0623	0.2496	-0.6746

Table 1: Results for Task 1 over the test data.

Experiment	Lang.	MAP@1/POT@1	MAP@3	MAP@5	MAP@10	Pot@3	Pot@5	Pot@10	Acc@1@tg1	Acc@2@tg1	Acc@3@tg1
LoRA Mistral7B, SC es dataset	es	0.6138	0.4124	0.2980	0.1595	0.7875	0.8246	0.8532	0.3288	0.4435	0.4839
Groq Prompting + CREA Freq	es	0.3136	0.2412	0.1650	0.089	0.5233	0.556	0.5893	0.1045	0.1905	0.2698
Baseline ALBERT + Distance	es	0.2563	0.154	0.1193	0.0731	0.4097	0.4907	0.5986	0.1079	0.1551	0.1854
LoRA Mistral7B, SC en-es dataset	en	0.5789	0.3718	0.2542	0.1355	0.7666	0.8087	0.8578	0.3438	0.4701	0.5526
LoRA Mistral7B, SC en-es-pt dataset	en	0.5947	0.3832	0.2634	0.1394	0.7824	0.828	0.8543	0.3789	0.5105	0.5701
Baseline ALBERT + Distance	en	0.1596	0.0920	0.0629	0.0379	0.2771	0.3438	0.4649	0.0824	0.1263	0.1561
LoRA Mistral7B, SC en-es dataset	pt	0.4021	0.2094	0.1360	0.0712	0.5784	0.6137	0.6631	0.2768	0.3844	0.4514
LoRA Mistral7B, SC en-es-pt dataset	pt	0.3756	0.2062	0.1336	0.0695	0.5414	0.5855	0.6172	0.2592	0.3562	0.4197
Baseline Static Embeddings + Word Freq	pt	0.0670	0.0380	0.0251	0.0136	0.1604	0.1922	0.2204	0.0582	0.0934	0.1358
LoRA Mistral7B, SC dataset en-es	all	0.3925	0.5233	0.5560	0.5893	0.2412	0.1650	0.0890	0.2156	0.2912	0.3324
LoRA Mistral7B, SC dataset en-es-pt	all	0.3818	0.2351	0.1608	0.0862	0.5091	0.5436	0.5772	0.2074	0.2851	0.3216

Table 2: Results for Task 2 over the test data.

ers, formatting this dataset to align with the task requirements. We fine-tuned the `Mistral-7B-v0.1` model, using a customized LoRA (Hu et al., 2022), choosing specific configurations to disable cache usage during training and to adapt the tokenizer for the corresponding task.

3.3 BERT and MLP models

As a totally different approach for task 1, we tried to use BERT embeddings as a text-representation input for Multilayer Perceptron models. For English we used the original BERT (Devlin et al., 2019), while for Spanish we used BETO (Cañete et al., 2020).

3.3.1 English (BERT)

To fine-tune the English BERT model we used the previously mentioned LCP2021 dataset. We trained for over 10 epochs with validation splits to monitor overfitting and batch processing for efficiency.

3.3.2 Spanish (BETO)

We had an additional problem when trying to fine-tune the BETO model, because there was not a Spanish dataset that was similar to LCP2021. The most similar set we found was the EASIER_CORPUS (Alarcon et al., 2023) dataset, but it only categorizes words in a binary way between easy and complex, and in this case we needed a more fine-grained distinction.

We first tried to generate synthetic text in Spanish using `gpt-3.5-turbo-0125`. In order to get data as balanced as possible, the prompts for the API were designed to produce sentences of two complexity levels, with a 50% probability each.

Then we gathered crowd-sourced data using a public website developed by us. This website allowed users to rate the complexity of words within sentences on a scale from 1 to 5. First we included only the synthetic sentences, and later on we also added the EASIER_CORPUS sentences, trying to include a wider range of linguistic contexts. We got approximately 2300 entries over a seven-day period[12].

After normalizing the scores of the whole dataset to match the expected score ranges, we fed BETO with all this Spanish text.

3.4 Use of pretrained models in Groq

As a final experiment for task 2 in Spanish, we used the Groq platform[13] to leverage the prompting capabilities of several pretrained LLMs: LLAMA (`llama2-70b-4096`), GEMMA (`gemma-7B-it`), and MIXTRAL (`mixtral-8x7B-32768`). We created a pipeline that prompts each of these models into giving simpler alternatives to a word in the context of a sentence, following a one-shot mechanism to illustrate the expected response. Using the Groq API, we collected the responses of the three models, combined them and used the word frequencies of the CREA corpus[14] to sort the possible answers.

[12] This manually annotated dataset will be published.

[13] https://groq.com/

[14] https://www.rae.es/banco-de-datos/crea

4 Results

In the appendix B we include tables 3 and 4, which show the results of our methods over the trial data. We used those preliminary results to choose which submissions to send to the competition, trying to keep the most promising systems but also a mix of different approaches. The experiments selected for submission are underlined in the tables. Tables 1 and 2 show the results of the submitted systems over the test set.

5 Conclusions

We presented a series of experiments for solving the Complex Word Prediction and Lexical Simplification tasks, ranging from simpler non-contextual static embeddings baselines, to more advanced fine-tuning of LLMs. The most important challenge in these tasks was the data scarcity, and because of this we had to use different resources like synthetic datasets, adapting existing datasets, or crowd-annotating new data. Our best approaches for both tasks where achieved by fine-tuning Mistral 7B, either with synthetic data or with already existing resources.

6 Limitations

Due to time constraints there were many experiments and combinations that we did not try, being the most salient one the fine-tuning of Mistral 7B with the manually annotated data collected through crowd-sourcing. We look forward to complete this experiment in the future.

References

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual nlp. *arXiv preprint arXiv:1307.1662*.

Rodrigo Alarcon, Lourdes Moreno, and Paloma Martínez. 2023. Easier corpus: A lexical simplification resource for people with cognitive impairments. *Plos one*, 18(4):e0283622.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

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. *Preprint*, arXiv:2005.14165.

José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. 2020. Spanish pre-trained bert model and evaluation data. In *PML4DC at ICLR 2020*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 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.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Kai North, Tharindu Ranasinghe, Matthew Shardlow, and Marcos Zampieri. 2024. Multils: A multi-task lexical simplification framework. *arXiv preprint arXiv:2402.14972*.

Gustavo Paetzold and Lucia Specia. 2016. 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).

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.

Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Huelsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Peréz Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, Sanja Štajner, Marcos Zampieri, and Horacio Saggion. 2024a. The BEA 2024 Shared Task on the

Multilingual Lexical Simplification Pipeline. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*.

Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Huelsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Peréz Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, Marcos Zampieri, and Horacio Saggion. 2024b. An Extensible Massively Multilingual Lexical Simplification Pipeline Dataset using the MultiLS Framework. In *Proceedings of the 3rd Workshop on Tools and Resources for People with REAding DIfficulties (READI)*.

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.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed 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*, volume 35, pages 24824–24837. Curran Associates, Inc.

A Synthetic data generation using Claude 3

In section 3.2.1 we briefly described the strategy to generate synthetic data, summarized by figure 1. In this appendix we will describe this pipeline in detail.

(1) Generation of synthetic sentences

A prompt was designed using four different sentence examples from the trial dataset as few-shot examples (Brown et al., 2020). To increase the diversity of the generated sentences and avoid overfitting, the examples are selected at random, so the prompt is not always the same. In some cases, to enhance variety, an additional phrase is added to the prompt asking the model to generate sentences containing at least one complex word.

The inference of the model is done multiple times with temperature or top-p set to 1 for maximizing diversity, and 250-500 sentences are created for each language, half of them with the complex word restriction added to the prompt.

(2) Selection of the target word

Given a generated sentence, we need to select a target word to generate (context, target) pairs, so we first tokenize the sentence to obtain a list of candidate words. Our simple tokenization means lower-casing, separating by spaces and removing punctuation and stopwords from NLTK (Bird and Loper, 2004)).

We explored two methods for selecting the words from the list of candadates. One approach was two select two or three words at random, not taking into account the complexity of the words. The other was to order the candidate words by decreasing complexity by prompting the LLM for this task, and then selecting the most complex and least complex words as target words.

(3) Generation of the complexity score and simpler alternatives

The prompt used to generate the complexity score and simpler alternatives for a given (context, target) pair consists of instructions for the model to generate the following three parts: a Chain-of-Thought analysis (CoT, Wei et al. (2022)) of the complexity of the target word in the given context sentence; a 1 to 5 complexity score for the target word, following the annotation guidelines used for the trial dataset; a list of at most three simpler alternatives for the target word. If no simpler alternatives exist, the model should return the same target word. Two handcrafted score examples are added to the prompt: one with a score of 1 and the other with a score of 5.

Each row of the resulting dataset consists of the context sentence, the target word, the analysis (CoT), the complexity score and the simpler alternatives. We elaborated a dataset of 2212 examples: 961 in Spanish, 750 in English and 500 in Portuguese.

Experiment	Lang.	Pearson	Spearman	MAE	MSE	RMSE	r2
BERT emeddings into MLP	en	0.3813	0.4331	0.2084	0.0543	0.2330	-0.3981
LoRA Mistral7B, LCP2021	en	0.8640	0.8574	0.1678	0.0330	0.1816	0.1514
MLP with RoBERTa embeddings	en	0.3957	0.2948	0.1607	0.0375	0.1936	0.0333
LoRA Mistral7B, SC en-es dataset	en	0.7363	0.7126	0.2243	0.0591	0.2431	-0.5199
MLP with BETO embeddings	es	0.4528	0.3925	0.2079	0.0622	0.2493	-0.1815
LoRA Mistral7B, es dataset	es	0.3892	0.3592	0.1942	0.0557	0.2360	-0.0570
LoRA Mistral7B, CoT es dataset	es	0.6355	0.6282	0.1458	0.0349	0.1867	0.3385
LoRA Mistral7B, SC es dataset	es	0.6461	0.6260	0.1483	0.0307	0.1754	0.4164
LoRA Mistral7B, SC-CoT es dataset	es	0.6102	0.6708	0.1575	0.0357	0.1890	0.3219
LoRA Mistral7B, SC en-es dataset	es	0.7283	0.7522	0.1337	0.0262	0.1618	0.5030
LoRA Mistral7B, SC en-es-pt dataset	es	0.7369	0.7180	0.1351	0.0259	0.1608	0.5090
LoRA Mistral7B, SC en-es-pt dataset	pt	0.7410	0.7754	0.1541	0.0415	0.2036	-0.5839
LoRA Mistral7B, SC en-es-pt dataset	ca	0.5460	0.5624	0.1299	0.0276	0.1662	-0.8219
LoRA Mistral7B, SC en-es-pt dataset	all	0.5301	0.5427	0.2060	0.0618	0.2486	-0.3930
Baseline Polyglot Embeddings + Word Freq	all	0.2106	0.2014	0.3711	0.2130	0.4615	-3.8008

Table 3: Results for Task 1 over the trial data. The underlined experiments are the ones we chose to send as submissions for the shared task.

Experiment	Lang.	MAP@1/POT@1	MAP@3	MAP@5	MAP@10	Pot@3	Pot@5	Pot@10	Acc@1@tg1	Acc@2@tg1	Acc@3@tg1
LoRA Mistral7B, es dataset	es	0.7666	0.5240	0.3144	0.1572	0.8666	0.8666	0.8666	0.5333	0.6000	0.6000
LoRA Mistral7B, CoT es dataset	es	0.6666	0.4722	0.2833	0.1416	0.8333	0.8333	0.8333	0.4000	0.4666	0.5333
LoRA Mistral7B, SC es dataset	es	0.9333	0.6000	0.4173	0.2298	0.9333	0.9666	1.000	0.5666	0.6666	0.7000
LoRA Mistral7B, SC-CoT es dataset	es	0.8000	0.5944	0.4510	0.2549	0.9000	0.9000	0.9333	0.4333	0.6333	0.6666
LoRA Mistral7B, SC en-es dataset	es	0.8666	0.5925	0.4405	0.2305	0.9333	0.9666	0.9666	0.5666	0.6333	0.7333
LoRA Mistral7B, SC en-es-pt dataset	es	0.8666	0.6333	0.4736	0.2658	0.9333	0.9333	0.9666	0.5000	0.6666	0.7333
Groq Prompting + CREA Freq	es	0.4666	0.2888	0.2213	0.1386	0.7000	0.7666	0.9333	0.2000	0.3666	0.4333
Baseline Static Embeddings + Distance	es	0.3333	0.1759	0.1355	0.0842	0.5000	0.6000	0.6666	0.1666	0.2333	0.2666
Baseline ALBERT + Distance	es	0.3333	0.2296	0.1714	0.1047	0.5333	0.5666	0.6333	0.1666	0.3000	0.3666
Baseline Static Embeddings + Word Freq	es	0.3000	0.2129	0.1511	0.0929	0.6000	0.6666	0.6666	0.1333	0.2333	0.3000
Baseline BERT + Distance	es	0.2666	0.2222	0.1810	0.1022	0.5000	0.5333	0.5666	0.1000	0.2333	0.2666
Baseline ALBERT + Word Freq	es	0.2000	0.1055	0.0730	0.0547	0.3666	0.4666	0.5666	0.1000	0.1666	0.2000
Baseline BERT + Word Freq	es	0.1333	0.0962	0.0677	0.0479	0.2333	0.3333	0.5333	0.0666	0.1000	0.1333
LoRA Mistral7B, SC en-es dataset	en	0.5666	0.3462	0.2371	0.1267	0.8333	0.8333	0.8333	0.4000	0.5666	0.7333
LoRA Mistral7B, SC en-es-pt dataset	en	0.5000	0.3166	0.2326	0.1201	0.7333	0.8333	0.8333	0.3666	0.5666	0.6333
Baseline Static Embeddings + Word Freq	en	0.1666	0.1074	0.0831	0.0480	0.3666	0.4666	0.5000	0.1000	0.2333	0.3000
Baseline BERT + Distance	en	0.1333	0.0981	0.0832	0.0462	0.3666	0.4666	0.5333	0.1000	0.2000	0.3000
Baseline Albert + Distance	en	0.1333	0.0814	0.0675	0.0387	0.3666	0.4666	0.5000	0.0666	0.1333	0.3000
Baseline Statitc Embeddings + Distance	en	0.0666	0.0574	0.0451	0.026	0.2333	0.3000	0.4000	0.0333	0.0666	0.2000
Baseline ALBERT + Word Freq	en	0.0666	0.0314	0.0245	0.0167	0.1333	0.2666	0.3666	0.0333	0.0666	0.1000
Baseline BERT + Word Freq	en	0.0333	0.0222	0.0183	0.0125	0.1333	0.1666	0.3666	0.000	0.000	0.0333
LoRA Mistral7B, SC en-es dataset	pt	0.3000	0.1925	0.1278	0.0695	0.6000	0.6666	0.7000	0.2333	0.4333	0.5000
LoRA Mistral7B, SC en-es-pt dataset	pt	0.3000	0.1759	0.1258	0.0646	0.6333	0.6333	0.6666	0.2333	0.3666	0.4333
Baseline Static Embeddings + Word Freq	pt	0.1333	0.0555	0.0333	0.0175	0.2000	0.2000	0.2666	0.1000	0.1333	0.1333
Baseline Static Embeddings + Distance	pt	0.0333	0.0166	0.0130	0.0078	0.0666	0.1333	0.2333	0.000	0.0333	0.0333
LoRA Mistral7B, SC dataset en-es	all	0.4066	0.2199	0.1319	0.0659	0.5300	0.5300	0.5300	0.2666	0.3333	0.3600
LoRA Mistral7B, SC dataset en-es-pt	all	0.4066	0.2257	0.1354	0.0677	0.4866	0.4866	0.4866	0.2466	0.3166	0.3566
Baseline Polyglot Embeddings + Word Freq	all	0.1133	0.0562	0.0384	0.022	0.1766	0.2133	0.2833	0.0500	0.0800	0.0866

Table 4: Results for Task 2 over the trial data. The underlined experiments are the ones we chose to send as submissions for the shared task.

B Results over the trial data

Tables 3 and 4 show the results of our methods over the trial data.

C Prompts

C.1 Generation of synthetic sentences prompt

System prompt: not used.

Message with role user:

```
Your task is to create new sentences in
{language}.

Here are some examples of the type of
sentences we expect:
{few_shot}

Try to write similar sentences to the
examples provided. {complex_sentence}
You should write {n} different and
diverse sentences, each in a new line.
No other text should be written.
```

Where:

1. **language** is the expected language of the sentences. For example: Spanish.

2. **few_shot** is a list of four examples of sentences from the trial dataset, separated by new-

lines.

3. **complex_sentence** is either an empty string or the sentence: `It is essential for the new sentences to use some extremely complex words.`
4. **n** is the amount of sentences the model should create in one run.

C.2 Order candidate words by complexity prompt

System prompt:

```
You are an annotator for a dataset of lexical simplification.

<task_description>
Given a context sentence and an a list of words from that context, your task is to order these words by decreasing complexity. The most complex word should go first, and the least complex word should go last.
</task_description>

<answer_format>
Your answer must follow the following format: Each word should be written in a new line. Nothing else should be written.
</answer_format>
```

Message with role user:

```
<context>
{context}
</context>

<words>
{words}
</words>
```

Where:

1. **context** is the sentence where the words appear.
2. **words** is the candidate words list separated by newlines.

C.3 Complexity score and simpler alternatives prompt

System prompt:

```
You are an annotator for a dataset of lexical simplification.

<task>
Given a context sentence and an a identified (whole-word) target to be evaluated, your task is to annotate the following information:
1) An step-by-step analysis of the target in the context to justify you following decisions.
2) A complexity score for the target in its context on a scale of 1 (easy) to 5 (difficult). This number should come as a consequence of the analyisis.
3) A list of no more than 3 simpler alternatives for the target, or the target itself if no simpler alternative can be found. The words should appear in increasing order of complexity. Do not add the target if simpler alternatives exist.
</task>

<considerations>
- The analysis should always have language learners in mind, not just native speakers.
- It is important to make decisions based on how other words could be evaluated, to make a grounded decision.
- If there are no simpler alternatives, the alternatives should only be the word itself.
</considerations>

<expected_answer>
Your answer must follow the following format:
- Inside XML tags <analysis></analisis> you must write (1) as free form text in english (regardless of source language). Remember to write in english.
- Inside XML tags <score></score> you must write (2) as one of the following numbers: 1, 2, 3, 4 or 5. Write only the number, without periods or text.
- Inside XML tags <simpler_alternatives> </simpler_alternatives> you must write (3) as a list of words separated by commas. No newlines between words should be used.
```

```
</expected_answer>

<score_examples>
Example of a score of 5:
 - Context: {example_context_1}
 - Target: {example_target_5}
Example of a score of 1:
 - Context: {example_context_1}
 - Target: {example_target_1}
</score_examples>
```

Message with role user:

```
<context>
{context}
</context>

<target>
{target}
</target>
```

Where:

1. **example_context_5** and **example_target_5** correspond to a hand-crafted score 5 example of a sentence and a target word respectively. Varies depending on the language.
2. **example_context_1** and **example_target_1** correspond to a hand-crafted score 1 example of a sentence and a target word respectively. Varies depending on the language.
3. **context** is the context sentence where the target word occurs.
4. **target** is the target word to evaluate.

C.4 Fine-tuning prompt format

The following is the prompt format used for the fine-tuning examples:

```
<context>
{context}
</context>
<target>
{target}
</target>
<score>
{score}
</score>
<simpler_alternatives>
{simpler_alternatives}
</simpler_alternatives>
```

Where:

1. **context** is the context sentence where the target word occurs.
2. **target** is the target word to evaluate.
3. **score** is a number between 1 and 5 corresponding to the complexity score.
4. **simple_alternatives** is a list of simpler alternatives for the target word, separated by commas.

GMU at MLSP 2024: Multilingual Lexical Simplification with Transformer Models

Dhiman Goswami, Kai North, Marcos Zampieri
George Mason University, USA
dgoswam@gmu.edu

Abstract

This paper presents GMU's submission to the Multilingual Lexical Simplification Pipeline (MLSP) shared task at the BEA workshop 2024. The task includes Lexical Complexity Prediction (LCP) and Lexical Simplification (LS) sub-tasks across 10 languages. Our submissions achieved rankings ranging from 1^{st} to 5^{th} in LCP and from 1^{st} to 3^{rd} in LS. Our best performing approach for LCP is a weighted ensemble based on Pearson correlation of language specific transformer models trained on all languages combined. For LS, GPT4-turbo zero-shot prompting achieved the best performance.

1 Introduction

Understanding LCP and LS is crucial for enhancing communication accessibility and readability across diverse linguistic contexts. LCP involves analyzing linguistic features to understand text difficulty, while LS focuses on making complex language more readable without losing its meaning. Therefore, LCP and LS provide inclusive communication and broadening access to information. Nowadays, NLP research is interested in identifying complex words which may be difficult for certain readers (Shardlow, 2013; Paetzold and Specia, 2016a). These difficult words requires various types of intervention, such as direct replacement in the setting of LS (Gooding and Kochmar, 2019), or generating further explanation (Rello et al., 2015)

Previously, the task of LCP involved labelling the complex words by binary classification (Paetzold and Specia, 2016a; Zampieri et al., 2017; Yimam et al., 2018). This approach was referred to as Complex Word Identification (CWI) which means a word can either be complex or not. However, in practice, word complexity should be a continuous value representing from the least to the most complex. Shardlow et al. (2021) and Shardlow et al. (2020b) were the first to introduce the task of LCP where a continuous value is assigned to identify a word's complexity. LS is the task of replacing difficult words with easier synonyms while preserving the information and intelligibility of the original text. This is a sub-task of Automatic Text Simplification (ATS) (Saggion and Hirst, 2017). Recently, similar to LCP, this task has also gained considerable amount of attention (Štajner et al., 2022).

In this paper, we use a cross-lingual weighted ensemble of transformer models to find LCP of a word in context of a sentence for 10 languages. For LS, we use GPT4-turbo (OpenAI, 2023) zero-shot prompting and also top 10 suggestions of GPT4-turbo and transformers models in terms of cosine similarity for 10 languages.

2 Related Work

2.1 Lexical Complexity Prediction

North et al. (2023c) is considered a comprehensive survey on LCP which provides us with a chronological journey of this task. LCP researchers traditionally used lexical features like word2vec, POS tag, frequency features including maximum entropy as traditional approaches (Paetzold and Specia, 2016a). Moreover, features like word length, frequency, n-gram features and word embeddings were also explored (Yimam et al., 2018) for LCP. On top of that, Binary classifiers such as SVMs, Decision Trees, Random Forests and threshold based metrics, variety of traditional machine learning classifiers and Neural Networks were used in different LCP systems. For example, the winning system CWI shared task of 2016 used a threshold-based methods and features extracted from Simple Wikipedia (Paetzold and Specia, 2016b) and Adaboost with WordNet features, POS tags, dependency parsing relations and psycholinguistic features were used by the winning system (Gooding and Kochmar, 2018) of BEA 2018.

From the approach of binary classification, LCP

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 627–634
June 20, 2024 ©2024 Association for Computational Linguistics

gradually shifted towards regression or probabilistic classification and thus transformer based models show better performance. A few years later, the idea of expressing complexity of words with a continuous value was first introduced on LCP shared task 2021 (Shardlow et al., 2021). A pre-trained transformer models fine-tuned for LCP (Pan et al., 2021) and a weighted ensemble of BERT and RoBERTa (Yaseen et al., 2021) respectively won the single word multi-word expressions sub-task of the shared task of 2021.

2.2 Lexical Simplification

LS research has utilized the word embedding models for retrieval or substitution generation (Glavaš and Štajner, 2015; Paetzold and Specia, 2016b). A pipeline of Substitute Generation (SG), Substitute Selection (SS) and Substitute Ranking (SR) was developed for this task. SG returns top-k most appropriate substitution of the complex word which are easy to understand and also preserve the original complex word's meaning and context. SS filters the generated top-k candidate substitutions and removes the unsuitable substitutions. SR orders the remaining top-k candidate substitutions by the decreasing order of simplicity and replace the complex word with the most suitable substitution (North et al., 2023b). Such approaches have proven better compared to earlier systems.

The state of the art for English LS was the LS-BERT system (Qiang et al., 2020) before 2022. It used a BERT (Devlin et al., 2018) based masking technique to find suitable simplifications for complex words and employed unsupervised ranking using various feature combinations. In 2022, Ferrés and Saggion (2022) introduced a benchmark dataset for LS in Spanish named ALEXSIS, and conducted experiments with various neural and unsupervised systems. They also evaluated an adaptation of LSBERT for Spanish, achieving state-of-the-art performance. Similarly, North et al. (2022b) developed and evaluated transformer models for Portuguese in 2022, based on a new corpus derived from ALEXSIS, following the BERT masked approach for substitute generation.

The first multilingual LS shared task was TSAR-2022 (Saggion et al., 2023). On this shared task, the best ranking for English was achieved using GPT-3 zero shot and few shot prompting (Aumiller and Gertz, 2023). For Portuguese, two customized pre-trained monolingual transformers and a large pre-trained monolingual model BERTimbau for masked language modeling achieved the best performance (North et al., 2022a). This prompting technique was further introduced in ALEXSIS+ (North et al., 2023a). Likewise, a masked language model followed by candidate token generation, candidate word selection and candidate word pruning along cosine similarity and parts of speech checking for substitution ranking (Whistely et al., 2022) was used for Spanish LS. Recently, a detailed Multi-task LS framework has been proposed by (North et al., 2024) which enables the creation of a multi-task LS dataset and training of a full LS pipeline.

3 Datasets

The MLSP shared-task (Shardlow et al., 2024a) covers 10 different languages - Catalan, English, Filipino, French, German, Italian, Japanese, Portuguese, Sinhala, Spanish and it has two sub-tasks- LCP and LS. LCP data instances include a sentence of a specific language and a specific word from that language of various text genre like news, religious, educational, Wikibooks etc. (Shardlow et al., 2024b). Then a complexity value ranging from 0-1 of that specific word in the context of that sentence is given. LS also has similar types of data instances but instead of a complexity value 10 simplified substitutions of the target word are provided for each instance. Moreover, MultiLS SP/CA dataset was used for both the LCP and LS task for Spanish and Catalan language (Bott et al., 2024). For each language, the data annotators are from different age group and professions like students, language learners, university faculty, freelancers. The data was annotated by both native and non-native speakers of each specific language. The data count for all the languages are shown in Table 1.

Language	Test
Catalan	445
English	570
Filipino	570
French	570
German	570
Italian	570
Japanese	570
Portuguese	569
Sinhala	600
Spanish	593
All Combined	5,627

Table 1: Data Distribution of Lexical Complexity Prediction and Lexical Simplification Dataset

There is no training data for this task. 30 Trial data was provided for each of the languages. For both the tasks we used all the trial data for validation. We performed cross-lingual transfer learning for the target language for LCP task. Moreover, only for the LCP task in English, we used CompLex dataset (Shardlow et al., 2020a) as training set for additional experiment. We merged 421 trial, 7,662 train and 917 test instances of this dataset and used these 9,000 instances together for the training purpose of English. We used the English trial data provided for this shared task as validation data in this case.

4 Experiments

Trial data provided for all the languages of the LCP task is very small. In general, it is common to use data augmentation and back-translation techniques to increase the number of data instances in such conditions (Akhbardeh et al., 2021). However, it will not work here as these techniques can change the word or even the context of the word after augmentation and back-translation causing change to the complexity also. As such, we use the idea of cross-lingual weighted ensemble approach by using trial data of all the languages for training and validation. We used 80-20 train and validation split. After that we use the test data of the target language for predicting lexical complexity. For training we have used weighted ensemble of mBERT (Devlin et al., 2018), XLM-R (Conneau et al., 2020) and language specific BERT models. For Catalan, Filipino, French, German, Italian, Japanese, Sinhala and Spanish we used calBERT (Codegram, 2020), RoBERTa-tagalog (Cruz and Cheng, 2021), flauBERT (Le et al., 2020), germanBERT (Dbmdz, 2020b), italianBERT (Dbmdz, 2020a), japaneseBERT (Tohoku-NLP, 2020), sinhalaBERTo (Dhananjaya et al., 2022) and spanishBERT (Cañete et al., 2020) respectively. For English we used BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019) and DeBERTa (He et al., 2021) as language specific model. For all language combined - ensemble of mBERT, XLM-R calBERT, DeBERTa, RoBERTa-tagalog, flauBERT, germanBERT, italianBERT, japaneseBERT, BERTimbau, sinhalaBERTo and spanishBERT were used. Pearson correlation coefficient was used as weight for the ensemble.

We use GPT4-turbo (Achiam et al., 2023) zero shot prompting which provides the best result for LS on both trial and test phase. Additionally, we used the same set up of BERT based models like LCP for all the languages to find the best 10 simplified substitutions for trial and test data. Then for each instances of a language, we took the set of all the words suggested by the BERT based models and GPT4-turbo together. After that, we find the embeddings of those words and the target token by LaBSE sentence transformer (Feng et al., 2020). Furthermore, we find the cosine similarity of the target token to the set of suggested word embeddings. Lastly, we choose the best 10 words by the decreasing order of cosine similarity of the embeddings.

5 Results

For LCP in English, we used the English trial data merged with the CompLex dataset and performed weighted ensemble. We rank 1^{st} with this procedure with Pearson correlation coefficient 0.8497. For the other 8 languages and all language combined we used the cross-lingual weighted ensemble. For Sinhala, we secure 3^{rd} rank with Pearson coefficient score 0.1246. For all language combined, Italian, Filipino, Spanish, Japanese, Catalan and German our rank is 4^{th} with Pearson coefficient 0.3494, 0.2919, 0.2823, 0.2438, 0.1775, 0.1549 and 0.1402 respectively. Lastly, we rank 5^{th} for French with 0.3193 Pearson coefficient. Test results for LCP are shown in Table 2.

For LS, zero-shot prompting by GPT-4 turbo performs the best for the 9 languages and all language combined. For Sinhala, we ranked 1^{st} with Accuracy@1@Top1 score 0.4182. For German, Spanish, all language combined, Japanese and Filipino - we stand 2^{nd} with Accuracy@1@Top1 0.42, 0.4182, 0.3345, 0.2583 and 0.0562 respectively. Lastly in the 3^{rd} position, we have English, Italian, French and Catalan with 0.5157, 0.4042, 0.3661 and 0.2247 Accuracy@1@Top1 respectively. The detailed explanation of the evaluation metrics used for LS is available at (Saggion et al., 2023). Test results for LS are shown in Table 2.

Trial results of LCP and LS are available in Table 4 and 5 of Appendix.

6 Error Analysis

For LCP the highest mean absolute and squared error are 0.2089 and 0.0589 for French and the lowest mean absolute and squared error are 0.1018 and 0.0168 for Sinhala. This is an acceptable mar-

	Test Scores (Target Language)				
Language	**Pearson**	**Spearman**	**MAE**	**MSE**	**R2**
Catalan	0.1549	0.1574	0.1462	0.0318	-0.3378
English (CompLex)	0.8497	0.7984	0.1137	0.0175	0.5247
Filipino	0.2823	0.2767	0.1164	0.0227	-0.0457
French	0.3193	0.3207	0.2089	0.0589	0.0484
German	0.1402	0.1473	0.1567	0.0413	-0.5279
Japanese	0.1775	0.1827	0.1363	0.0270	0.0241
Sinhala	0.1246	0.1303	0.1018	0.0168	-0.0370
Spanish	0.2438	0.1984	0.1630	0.0379	-0.0731
All Combined	0.3494	0.3642	0.1464	0.0331	0.1094

Table 2: Test Results of LCP (Weighted Ensemble of the Models Used for Corresponding Languages in Trial Phase)

Language	Models	A@1@Top1	A@2@Top1	A@3@Top1	MacAvgPrec@1	MacAvgPrec@3	MacAvgPrec@5	MacAvgPrec@10	MAP@3	MAP@5	MAP@10
Catalan	GPT4-turbo	**0.2247**	**0.3056**	**0.328**	**0.537**	**0.7101**	**0.7573**	**0.8044**	**0.362**	**0.2641**	**0.1582**
	Top10Suggestion	0.0651	0.1191	0.1595	0.2426	0.5191	0.6404	0.755	0.172	0.1408	0.0893
English	GPT4-turbo	**0.5157**	**0.635**	**0.6894**	**0.7491**	**0.8754**	**0.907**	**0.928**	**0.513**	**0.3691**	**0.2095**
	Top10Suggestion	0.1929	0.3228	0.4157	0.335	0.6315	0.7649	0.8649	0.2339	0.1869	0.1106
Filipino	GPT4-turbo	**0.0562**	**0.0632**	**0.0685**	**0.2934**	**0.3989**	**0.4358**	**0.4868**	**0.1395**	**0.0916**	**0.0491**
	Top10Suggestion	0.0157	0.0228	0.0245	0.0807	0.1842	0.2859	0.3859	0.0449	0.0338	0.0201
French	GPT4-turbo	**0.3661**	**0.4559**	**0.514**	**0.7411**	**0.8679**	**0.889**	**0.9154**	**0.5148**	**0.3946**	**0.2447**
	Top10Suggestion	0.0845	0.1672	0.2394	0.2271	0.5316	0.6971	0.8257	0.1725	0.149	0.1023
German	GPT4-turbo	**0.42**	**0.5043**	**0.5817**	**0.6414**	**0.7908**	**0.8312**	**0.8558**	**0.4002**	**0.2874**	**0.1631**
	Top10Suggestion	0.1192	0.2228	0.3	0.2578	0.5491	0.6666	0.7982	0.1852	0.1463	0.092
Italian	GPT4-turbo	**0.4042**	**0.5641**	**0.6309**	**0.7346**	**0.8822**	**0.9244**	**0.9402**	**0.4615**	**0.3328**	**0.1966**
	Top10Suggestion	0.1546	0.2724	0.3567	0.3567	0.6625	0.7855	0.8717	0.246	0.1965	0.1242
Japanese	GPT4-turbo	**0.2583**	**0.3708**	**0.4393**	**0.5413**	**0.6801**	**0.7223**	**0.7627**	**0.3618**	**0.2599**	**0.1529**
	Top10Suggestion	0.1195	0.2144	0.2847	0.3075	0.5817	0.6731	0.7469	0.2144	0.171	0.1107
Sinhala	GPT4-turbo	**0.2284**	**0.2829**	**0.3163**	**0.311**	**0.4165**	**0.4815**	**0.536**	**0.1387**	**0.0894**	**0.0469**
	Top10Suggestion	0.13	0.2372	0.3057	0.195	0.3848	0.4639	0.5272	0.1147	0.0759	0.0394
Spanish	GPT4-turbo	**0.4182**	**0.5362**	**0.6087**	**0.801**	**0.9173**	**0.9477**	**0.9612**	**0.5987**	**0.4653**	**0.2853**
	Top10Suggestion	0.236	0.3558	0.4704	0.5919	0.86	0.9106	0.9392	0.4371	0.3542	0.2244
All Combined	GPT4-turbo	**0.3345**	**0.4291**	**0.4828**	**0.5934**	**0.7276**	**0.7695**	**0.803**	**0.379**	**0.2754**	**0.1614**
	Top10Suggestion	0.1331	0.2261	0.2999	0.2876	0.5374	0.6467	0.7386	0.1981	0.1561	0.0971

Table 3: Test Results of LS (Top 10 Suggestions are Selected from the Output of GPT4-turbo and the Models Used for Corresponding Languages in Trial Phase)

gin of error when we are training a model with cross-lingual data and testing with language specific data. This is also a reason of getting negative R2 score for 4 languages which testifies that the data struggles to fit the regression model for those languages.

For LS, zero-shot prompting by GPT4-turbo alone provides the best result but when we try to find the best 10 suggestions from the set of suggestions generated by the BERT based models and GPT4-turbo together, the result significantly decreases. This was because the target token in the sentence varied be in different grammatical form. Therefore, finding proper simplified suggestions that fits the context proves to be a struggle for the BERT based model.

7 Conclusion

Our team *GMU*'s approaches in MLSP 2024 shared task achieved competitive results across multiple languages for both the LCP and LS sub-tasks. The weighted ensemble technique based on transformer models proved effective for LCP, while GPT-4 zero-shot prompting excelled at LS. The multilingual nature of this shared task also highlights the importance of developing techniques that can generalize across languages.

One key limitation of our approach is the reliance on cross-lingual transfer due to limited language-specific training data for most languages. While this allowed sharing resources across languages, having larger datasets to each language could potentially boost performance. Additionally, the error analysis revealed some remaining challenges in handling complex word expressions and phrases during LS. Further improvements in modeling could address these cases more effectively for MLSP in future.

Acknowledgements

We express our gratitude to the organizers of the BEA 2024 workshop for facilitating this insightful shared task and providing the multilingual datasets that enabled our research. We also thank the annotators for successfully annotating this large dataset.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Farhad Akhbardeh, Cecilia Ovesdotter Alm, Marcos Zampieri, and Travis Desell. 2021. Handling extreme class imbalance in technical logbook datasets. In *Proceedings of ACL (IJCNLP)*.

Dennis Aumiller and Michael Gertz. 2023. Unihd at tsar-2022 shared task: Is compute all we need for lexical simplification? *arXiv preprint arXiv:2301.01764*.

Stefan Bott, Horacio Saggion, Nelson Peréz Rojas, Martin Solis Salazar, and Saul Calderon Ramirez. 2024. Multils-sp/ca: Lexical complexity prediction and lexical simplification resources for catalan and spanish.

José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. 2020. Spanish pre-trained bert model and evaluation data. In *Proceedings of PML4DC (ICLR)*.

Codegram. 2020. Calbert: A catalan language model. https://huggingface.co/codegram/calbert-base-uncased.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL*.

Jan Christian Blaise Cruz and Charibeth Cheng. 2021. Improving large-scale language models and resources for filipino. *arXiv preprint arXiv:2111.06053*.

Dbmdz. 2020a. BERT-base italian cased model. https://huggingface.co/dbmdz/bert-base-italian-cased.

Dbmdz. 2020b. German bert model. https://huggingface.co/dbmdz/bert-base-german-uncased.

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*.

Vinura Dhananjaya, Piyumal Demotte, Surangika Ranathunga, and Sanath Jayasena. 2022. Bertifying sinhala–a comprehensive analysis of pre-trained language models for sinhala text classification. *arXiv preprint arXiv:2208.07864*.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

Daniel Ferrés and Horacio Saggion. 2022. Alexsis: A dataset for lexical simplification in spanish. In *Proceedings of LREC*.

Goran Glavaš and Sanja Štajner. 2015. Simplifying lexical simplification: Do we need simplified corpora? In *Proceedings of ACL-IJCNLP*.

Sian Gooding and Ekaterina Kochmar. 2018. Camb at cwi shared task 2018: Complex word identification with ensemble-based voting. In *Proceedings of BEA*.

Sian Gooding and Ekaterina Kochmar. 2019. Recursive context-aware lexical simplification. In *Proceedings of EMNLP-IJCNLP*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *Proceedings of ICLR*.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. 2020. Flaubert: des modèles de langue contextualisés pré-entraînés pour le français. In *Proceedings of TALN*.

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.

Kai North, Alphaeus Dmonte, Tharindu Ranasinghe, Matthew Shardlow, and Marcos Zampieri. 2023a. Alexsis+: Improving substitute generation and selection for lexical simplification with information retrieval. In *Proceedings of BEA*.

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. 2023b. Deep learning approaches to lexical simplification: A survey. *arXiv preprint arXiv:2305.12000*.

Kai North, Tharindu Ranasinghe, Matthew Shardlow, and Marcos Zampieri. 2024. Multils: A multi-task lexical simplification framework. *arXiv preprint arXiv:2402.14972*.

Kai North, Marcos Zampieri, and Tharindu Ranasinghe. 2022b. Alexsis-pt: A new resource for portuguese lexical simplification. *arXiv preprint arXiv:2209.09034*.

Kai North, Marcos Zampieri, and Matthew Shardlow. 2023c. Lexical complexity prediction: An overview. *ACM Computing Surveys*, 55(9):1–42.

OpenAI. 2023. GPT-4 technical report. https://arxiv.org/abs/2303.08774.

Gustavo Paetzold and Lucia Specia. 2016a. Semeval 2016 task 11: Complex word identification. In *Proceedings of SemEval*, pages 560–569.

Gustavo Paetzold and Lucia Specia. 2016b. Sv000gg at semeval-2016 task 11: Heavy gauge complex word identification with system voting. In *Proceedings of SemEval*.

Chunguang Pan, Bingyan Song, Shengguang Wang, and Zhipeng Luo. 2021. Deepblueai at semeval-2021 task 1: Lexical complexity prediction with a deep ensemble approach. In *Proceedings of SemEval*.

Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, and Xindong Wu. 2020. Lexical simplification with pretrained encoders. In *Proceedings of the AAAI*.

Luz Rello, Roberto Carlini, Ricardo Baeza-Yates, and Jeffrey P Bigham. 2015. A plug-in to aid online reading in spanish. In *Proceedings of W4A*.

Horacio Saggion and Graeme Hirst. 2017. *Automatic text simplification*. Springer.

Horacio Saggion, Sanja Štajner, Daniel Ferrés, Kim Cheng Sheang, Matthew Shardlow, Kai North, and Marcos Zampieri. 2023. Findings of the tsar-2022 shared task on multilingual lexical simplification. *arXiv preprint arXiv:2302.02888*.

Matthew Shardlow. 2013. A comparison of techniques to automatically identify complex words. In *Proceedings of SRW*.

Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Huelsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Peréz Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, Sanja Štajner, Marcos Zampieri, and Horacio Saggion. 2024a. The BEA 2024 Shared Task on the Multilingual Lexical Simplification Pipeline. In *Proceedings of BEA*.

Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Huelsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Peréz Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, Marcos Zampieri, and Horacio Saggion. 2024b. An Extensible Massively Multilingual Lexical Simplification Pipeline Dataset using the MultiLS Framework. In *Proceedings of READI*.

Matthew Shardlow, Michael Cooper, and Marcos Zampieri. 2020a. CompLex — a new corpus for lexical complexity prediction from Likert Scale data. In *Proceedings of READI*.

Matthew Shardlow, Michael Cooper, and Marcos Zampieri. 2020b. Complex: A new corpus for lexical complexity prediction from likert scale data. *arXiv preprint arXiv:2003.07008*.

Matthew Shardlow, Richard Evans, Gustavo Henrique Paetzold, and Marcos Zampieri. 2021. Semeval-2021 task 1: Lexical complexity prediction. *arXiv preprint arXiv:2106.00473*.

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*.

Tohoku-NLP. 2020. BERT-base japanese model. https://huggingface.co/tohoku-nlp/bert-base-japanese.

Peniel 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*.

Tuqa Bani Yaseen, Qusai Ismail, Sarah Al-Omari, Eslam Al-Sobh, and Malak Abdullah. 2021. Just-blue at semeval-2021 task 1: Predicting lexical complexity using bert and roberta pre-trained language models. In *Proceedings of SemEval*.

Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo H Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri. 2018. A report on the complex word identification shared task 2018. *arXiv preprint arXiv:1804.09132*.

Marcos Zampieri, Shervin Malmasi, Gustavo Paetzold, and Lucia Specia. 2017. Complex word identification: Challenges in data annotation and system performance. In *Proceedings of NLPTEA*.

A Appendix

		Validation Scores (Combined Dataset)				
Language	**Models**	**Pearson**	**Spearman**	**MAE**	**MSE**	**R2**
Catalan	mBERT	0.5839	0.5965	0.1296	0.0272	0.2676
	XLM-R	0.4131	0.3881	0.1496	0.0333	0.1051
	calBERT	0.4724	0.4868	0.1384	0.031	0.1653
English (All Trial)	mBERT	0.5926	0.57	0.1422	0.0297	0.2012
	XLM-R	0.4245	0.4345	0.1496	0.0335	0.0982
	BERT	0.4396	0.4489	0.1593	0.0349	0.0621
	RoBERTa	0.5418	0.5525	0.1375	0.0266	0.2848
	DeBERTa	0.5437	0.5234	0.138	0.0276	0.257
English (CompLex)	BERT	0.7732	0.751	0.1478	0.0288	0.2604
	RoBERTa	0.6454	0.7072	0.156	0.0325	0.1635
	DeBERTa	0.8144	0.7434	0.1486	0.0269	0.3094
Filipino	mBERT	0.5814	0.577	0.1299	0.027	0.2744
	XLM-R	0.4447	0.4368	0.1453	0.031	0.1665
	RoBERTa-tagalog	0.4162	0.3686	0.1504	0.0342	0.0807
French	mBERT	0.5703	0.6264	0.1402	0.027	0.2734
	XLM-R	0.4588	0.4576	0.1466	0.0306	0.1778
	flauBERT	0.3742	0.3068	0.1485	0.0322	0.1345
German	mBERT	0.6061	0.6159	0.1382	0.0263	0.29933
	XLM-R	0.4586	0.4481	0.1469	0.0296	0.2043
	germanBERT	0.4511	0.4669	0.1415	0.0306	0.1778
Italian	mBERT	0.6196	0.5757	0.1225	0.0244	0.3441
	XLM-R	0.4934	0.4625	0.144	0.0297	0.2003
	italianBERT	0.5577	0.5419	0.1353	0.0262	0.2946
Japanese	mBERT	0.5551	0.5568	0.1378	0.0301	0.1914
	XLM-R	0.5479	0.5355	0.1422	0.028	0.2462
	japaneseBERT	0.4286	0.4285	0.1521	0.0341	0.083
Sinhala	mBERT	0.5948	0.6375	0.1333	0.0263	0.2929
	XLM-R	0.4396	0.4569	0.1414	0.0304	0.181
	sinhalaBERTo	0.3766	0.4027	0.1568	0.0337	0.0923
Spanish	mBERT	0.5412	0.5861	0.136	0.0282	0.2428
	XLM-R	0.5119	0.5022	0.1391	0.0289	0.2225
	spanishBERT	0.4141	0.3909	0.1559	0.0328	0.1188
All Combined	mBERT	0.4511	0.495	0.1546	0.0326	0.1223
	XLM-R	0.4588	0.4576	0.1466	0.0306	0.1778
	calBERT	0.4044	0.4069	0.1517	0.0326	0.1226
	DeBERTa	0.4511	0.4745	0.1482	0.0318	0.1454
	RoBERTa-tagalog	0.4626	0.4588	0.1564	0.0354	0.0489
	flauBERT	0.4416	0.4236	0.1583	0.036	0.0306
	germanBERT	0.4383	0.4261	0.1531	0.0345	0.718
	italianBERT	0.5577	0.5419	0.1353	0.0262	0.2946
	japaneseBERT	0.4183	0.4461	0.1543	0.0347	0.0675
	BERTimbau	0.5274	0.5701	0.1306	0.0271	0.2697
	sinhalaBERTo	0.4249	0.4622	0.156	0.0338	0.0919
	spanishBERT	0.4679	0.4499	0.1535	0.0356	0.0433

Table 4: Trial Results of LCP

Language	Models	A@1@Top1	A@2@Top1	A@3@Top1	MacAvgPrec@1	MacAvgPrec@3	MacAvgPrec@5	MacAvgPrec@10	MAP@3	MAP@5	MAP@10
Catalan	mBERT	0.0666	0.1333	0.1333	0.1	0.1666	0.1666	0.2	0.0999	0.0866	0.0437
	XLM-R	0.0066	0.2333	0.3	0.1333	0.3666	0.4	0.4	0.1351	0.1094	0.0547
	calBERT	0.0666	0.1666	0.1666	0.1	0.1666	0.2333	0.2333	0.0999	0.0893	0.0446
	GPT4-turbo	**0.4666**	**0.4666**	**0.5**	**0.4666**	**0.6**	**0.7333**	**0.7333**	**0.2407**	**0.1634**	**0.0888**
	Top10Suggestion	0.2	0.2666	0.3333	0.2333	0.5	0.5666	0.6666	0.1259	0.0932	0.0524
English	mBERT	0.1	0.2	0.26	0.2	0.4666	0.5	0.6333	0.1481	0.1012	0.0577
	XLM-R	0.1	0.1666	0.2666	0.1666	0.4333	0.5666	0.6333	0.1222	0.0796	0.0484
	BERT	0.1666	0.1666	0.2	0.3	0.5333	0.5666	0.7	0.174	0.1297	0.0766
	RoBERTa	0.066	0.2	0.2333	0.2	0.4666	0.6333	0.7333	0.1648	0.1335	0.0815
	DeBERTa	0.1666	0.1666	0.1666	0.2333	0.2333	0.2333	0.2333	0.2	0.1446	0.0733
	GPT4-turbo	**0.4**	**0.5**	**0.5666**	**0.7**	**0.8**	**0.8666**	**0.8666**	**0.4444**	**0.3136**	**0.1728**
	Top10Suggestion	0.1333	0.2666	0.3666	0.2666	0.6666	0.6666	0.6666	0.174	0.1224	0.0612
Filipino	mBERT	0.0333	0.0666	0.0666	0.0333	0.0666	0.0666	0.1	0.0166	0.01	0.0054
	XLM-R	0.1333	0.2	0.2	0.1333	0.2	0.2	0.2333	0.0888	0.0533	0.0271
	RoBERTa-tagalog	0.2	0.2666	0.3	0.2333	0.3333	0.4	0.4333	0.1037	0.0652	0.0352
	GPT4-turbo	**0.3666**	**0.3666**	**0.3666**	**0.4**	**0.4333**	**0.4666**	**0.5**	**0.1611**	**0.1053**	**0.055**
	Top10Suggestion	0.0666	0.1333	0.2333	0.0666	0.2333	0.3333	0.4	0.0555	0.0373	0.0206
French	mBERT	0.2	0.3333	0.4	0.2666	0.4333	0.5	0.5	0.1611	0.0996	0.052
	XLM-R	0.1666	0.3	0.3666	0.2333	0.4333	0.5	0.5333	0.1185	0.0711	0.0402
	flauBERT	0.0166	0.0266	0.0366	0.0166	0.0266	0.0366	0.0366	0.0107	0.0071	0.0046
	GPT4-turbo	**0.5**	**0.6333**	**0.6666**	**0.7**	**0.8**	**0.8**	**0.8**	**0.3759**	**0.2305**	**0.1169**
	Top10Suggestion	0.2	0.2333	0.2333	0.2333	0.4333	0.5666	0.7333	0.1296	0.0927	0.0518
German	mBERT	0.0333	0.0666	0.0666	0.0333	0.0666	0.0666	0.1	0.0287	0.019	0.0111
	XLM-R	0.0333	0.0666	0.1333	0.1	0.1666	0.3	0.3333	0.0446	0.03	0.0168
	germanBERT	0.1666	0.2	0.2333	0.1333	0.2333	0.2333	0.2333	0.0814	0.0592	0.0299
	GPT4-turbo	**0.6**	**0.8666**	**0.9666**	**0.7333**	**0.9**	**0.9**	**0.9**	**0.3944**	**0.2603**	**0.137**
	Top10Suggestion	0.0333	0.0666	0.1666	0.0666	0.2666	0.4333	0.7	0.0555	0.053	0.0337
Italian	mBERT	0.0333	0.0666	0.0666	0.0333	0.1	0.1333	0.2	0.0222	0.015	0.0092
	XLM-R	0.1	0.1	0.1	0.1333	0.1333	0.1666	0.2333	0.0444	0.028	0.0158
	italianBERT	0.2333	0.3666	0.4	0.2666	0.4666	0.5333	0.6	0.1537	0.1038	0.0518
	GPT4-turbo	**0.5**	**0.6**	**0.7**	**0.3518**	**0.2334**	**0.1267**	**0.6**	**0.3518**	**0.2334**	**0.1267**
	Top10Suggestion	0.1666	0.2	0.2333	0.2	0.3666	0.5666	0.7666	0.1259	0.0905	0.0566
Japanese	mBERT	0.0666	0.0666	0.0666	0.0666	0.0666	0.0666	0.0666	0.0518	0.0427	0.0213
	XLM-R	0.0666	0.0666	0.0666	0.0666	0.0666	0.0666	0.0666	0.0518	0.0427	0.0213
	japaneseBERT	0.1	0.1333	0.1666	0.1333	0.1666	0.1666	0.1666	0.137	0.0955	0.0477
	GPT4-turbo	**0.4333**	**0.4666**	**0.4666**	**0.5333**	**0.6333**	**0.7333**	**0.8**	**0.2629**	**0.1767**	**0.0936**
	Top10Suggestion	0.0333	0.0666	0.1	0.0666	0.1333	0.2333	0.5	0.0407	0.0321	0.0226
Sinhala	mBERT	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0833	0.0599	0.0299
	XLM-R	0.0333	0.0666	0.0666	0.1	0.0133	0.0133	0.0233	0.0481	0.0288	0.0159
	sinhalaBERTo	0.0333	0.0333	0.0333	0.0333	0.0333	0.0333	0.0333	0.0111	0.0066	0.0033
	GPT4-turbo	**0.3666**	**0.5**	**0.5666**	**0.5333**	**0.7333**	**0.7666**	**0.8333**	**0.2851**	**0.1757**	**0.0961**
	Top10Suggestion	0.2	0.3333	0.4	0.3666	0.5666	0.6	0.7666	0.2037	0.1412	0.0786
Spanish	mBERT	0.1	0.1333	0.1333	0.1333	0.1333	0.1333	0.1333	0.1018	0.0744	0.0418
	XLM-R	0.1666	0.3	0.3666	0.2333	0.4333	0.5	0.5333	0.1185	0.0711	0.0402
	spanishBERT	0.2666	0.3333	0.4333	0.3333	0.5666	0.6666	0.7333	0.2055	0.133	0.0698
	GPT4-turbo	**0.4**	**0.6333**	**0.7666**	**0.6333**	**0.8666**	**0.9333**	**0.9333**	**0.4018**	**0.2721**	**0.1433**
	Top10Suggestion	0.2666	0.3333	0.4666	0.3	0.6333	0.7333	0.7666	0.1888	0.132	0.0716
All Combined	mBERT	0.0233	0.04	0.0566	0.0433	0.1	0.11	0.1533	0.0287	0.019	0.0111
	XLM-R	0.0466	0.0833	0.1033	0.0866	0.1533	0.2033	0.2366	0.0446	0.03	0.0168
	calBERT	0.0333	0.0366	0.0366	0.0333	0.0366	0.0366	0.0366	0.03	0.02	0.01
	DeBERTa	0.0333	0.0366	0.0366	0.0333	0.0366	0.0366	0.0366	0.03	0.02	0.01
	flauBERT	0.0166	0.0266	0.0366	0.0166	0.0266	0.0366	0.0366	0.0107	0.0071	0.0046
	germanBERT	0.0166	0.02	0.0233	0.02	0.03	0.0333	0.0333	0.0081	0.0056	0.0028
	italianBERT	0.0266	0.04	0.0433	0.0333	0.0533	0.0666	0.0766	0.0175	0.0118	0.006
	japaneseBERT	0.0333	0.0366	0.04	0.0366	0.04	0.04	0.04	0.0303	0.0202	0.0101
	BERTimbau	0.0066	0.0066	0.0066	0.0066	0.0066	0.0066	0.0066	0.0022	0.0013	0.0006
	sinhalaBERTo	0.0033	0.0033	0.0033	0.0166	0.03	0.04	0.0533	0.0083	0.0062	0.0035
	spanishBERT	0.0033	0.0033	0.0033	0.0066	0.0166	0.02	0.03	0.0033	0.0022	0.0012
	GPT4-turbo	**0.39**	**0.48**	**0.5333**	**0.5966**	**0.7433**	**0.7933**	**0.8366**	**0.3122**	**0.2088**	**0.1111**
	Top10Suggestion	0.1166	0.2166	0.2933	0.1833	0.45	0.5833	0.6833	0.1248	0.0942	0.0526

Table 5: Trial Results of LS

ITEC at MLSP 2024:
Transferring Predictions of Lexical Difficulty from Non-Native Readers

Anaïs Tack
KU Leuven
Faculty of Arts, Research Unit Linguistics
imec research group itec
anais.tack@kuleuven.be

Abstract

This paper presents the results of our team's participation in the BEA 2024 shared task on the multilingual lexical simplification pipeline (MLSP; Shardlow et al., 2024a). During the task, organizers supplied data that combined two components of the simplification pipeline: lexical complexity prediction and lexical substitution. This dataset encompassed ten languages, including French. Given the absence of dedicated training data, teams were challenged with employing systems trained on pre-existing resources and evaluating their performance on unexplored test data.

Our team contributed to the task using previously developed models for predicting lexical difficulty in French (Tack, 2021). These models were built on deep learning architectures, adding to our participation in the CWI 2018 shared task (De Hertog and Tack, 2018). The training dataset comprised 262,054 binary decision annotations, capturing perceived lexical difficulty, collected from a sample of 56 non-native French readers. Two pre-trained neural logistic models were used: (1) a model for predicting difficulty for words within their sentence context, and (2) a model for predicting difficulty for isolated words.

The findings revealed that despite being trained for a distinct prediction task (as indicated by a negative R^2 fit), transferring the logistic predictions of lexical difficulty to continuous scores of lexical complexity exhibited a positive correlation. Specifically, the results indicated that isolated predictions exhibited a higher correlation ($r = .36$) compared to contextualized predictions ($r = .33$). Moreover, isolated predictions demonstrated a remarkably higher Spearman rank correlation ($\rho = .50$) than contextualized predictions ($\rho = .35$). These results align with earlier observations by Tack (2021), suggesting that the ground truth primarily captures more lexical access difficulties than word-to-context integration problems.

1 Introduction

The aim of predicting and simplifying lexical difficulty is to enhance text readability by focusing on vocabulary. Drawing from a simplified perspective on reading (Hoover and Gough, 1990), we can divide these difficulties into two main categories: decoding and comprehension. Decoding issues relate to difficulties in accessing words (also known as "lexical access"), where readers struggle to recognize and recall the form and meaning of words from memory. Conversely, comprehension difficulties involve struggles in integrating words into the broader textual context (sometimes termed "word-to-context integration"). Therefore, simplifying lexical difficulty entails employing various strategies to boost clarity and comprehension. This may involve substituting complex terms with simpler alternatives or providing contextual clues or definitions. Ultimately, the goal is to enhance accessibility while maintaining the integrity of the conveyed message.

Over the last decade, several tasks have been organized to advance the development of automated models, including the complex word identification shared task (Paetzold and Specia, 2016), the second complex word identification shared task (Yimam et al., 2018), the shared task on lexical complexity prediction (Shardlow et al., 2021), and the shared task on multilingual lexical simplification (Saggion et al., 2022). Lastly, Shardlow et al. (2024a) organized the shared task on multilingual lexical simplification pipeline (MLSP).[1]

This system description paper outlines our team's involvement in the MLSP shared task, focusing on our approach. Specifically, we leveraged predictions of lexical difficulty for French from previous research (Tack, 2021) in the initial phase of the lexical simplification pipeline, known

[1] https://sites.google.com/view/mlsp-sharedtask-2024/

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 635–639
June 20, 2024 ©2024 Association for Computational Linguistics

as lexical complexity prediction. Our approach also entailed comparing predictions for individual words (approximating lexical access difficulties) with predictions for words within context (approximating word-to-context integration difficulties). Subsequent sections detail our methodology and findings.

2 Method

The shared task progressed through two distinct phases. In the development phase, which took place from February 15 to March 14, 2024, teams were tasked with developing systems using existing resources. Due to the absence of dedicated training data and the small size of only 30 trial items per language, our emphasis was on employing pre-trained models for making zero-shot predictions of lexical difficulty (see Section 2.1).

During the evaluation phase, from March 15 to March 26, 2024, teams were provided with test data for ten languages within the MultiLS framework (Shardlow et al., 2024b; North et al., 2024). During this phase, we used our pre-trained models to predict scores of lexical complexity for the French test set (see Section 2.2) and made two submissions. In the subsequent sections, we will provide a more detailed description of the pre-trained models and test data.

2.1 Pre-Trained Models for French

We employed two neural models for predicting lexical difficulty in French, previously developed by Tack (2021) in her Ph.D. thesis. These models represented an improved version of the deep learning architecture developed by De Hertog and Tack (2018) for the second shared task on complex word identification (Yimam et al., 2018) and the earlier models developed by Tack et al. (2016b).

The first model featured a bidirectional long short-term memory neural network architecture, depicted in Figure 1. This model, constructed using TensorFlow, incorporated two word representations as input: character embeddings (generated through a convolutional neural network) and pre-trained FastText word embeddings. Furthermore, the model integrated learner-specific encodings to tailor predictions accordingly. However, in the transfer approach, personalization was not possible, resulting in these encodings being set to zero for the shared task.

The second model comprised a feedforward neu-

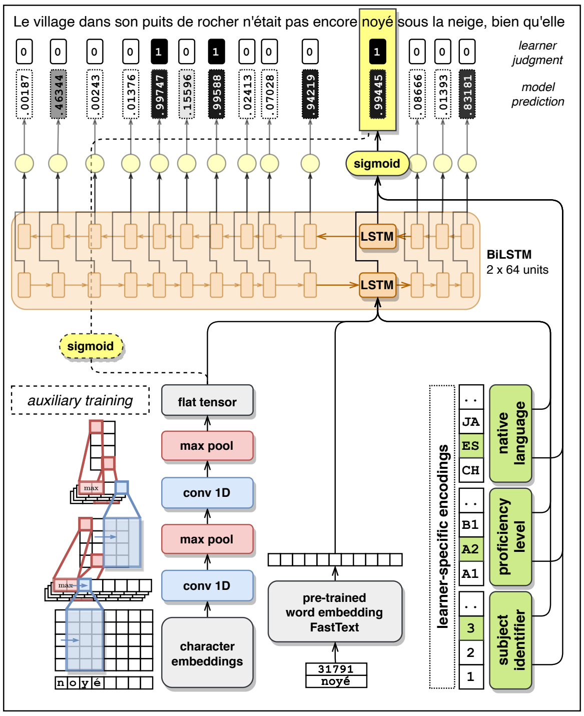


Figure 1: Bidirectional Long-Short Term Memory Neural Network Architecture in Tack (2021), Making Contextualized Predictions of Lexical Difficulty for French

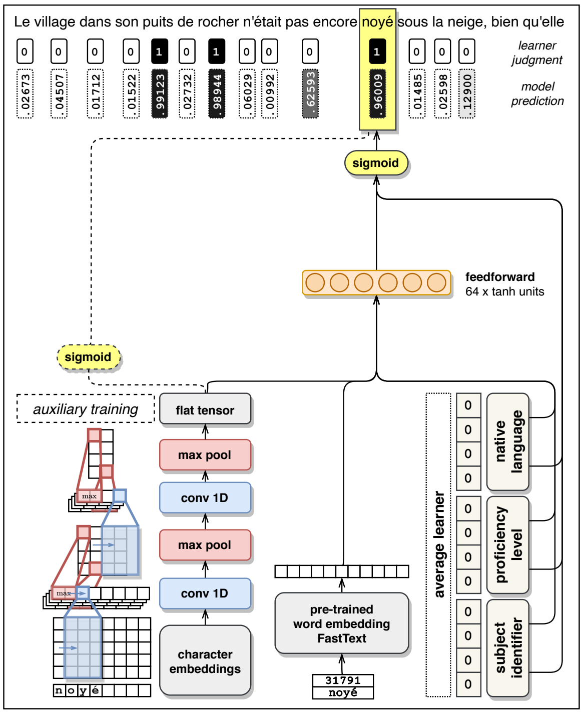


Figure 2: Feedforward Neural Network Architecture in Tack (2021), Making Isolated Predictions of Lexical Difficulty for French

ID	Language	Sentence Context	Target Word
fr_549	french	Bien sûr, on peut me rétorquer que je n'ai qu'à acquérir la nationalité française.	rétorquer
fr_550	french	Bien sûr, on peut me rétorquer que je n'ai qu'à acquérir la nationalité française.	acquérir
fr_551	french	Bien sûr, on peut me rétorquer que je n'ai qu'à acquérir la nationalité française.	nationalité

Figure 3: Examples of Items in the French Test Data

ral network architecture, as depicted in Figure 2. Built using TensorFlow, this model utilized two word representations as input: character embeddings (generated through a convolutional neural network) and pre-trained FastText word embeddings. Additionally, learner-specific encodings were incorporated into the model to customize predictions. However, in the transfer approach where personalization wasn't possible, these encodings were also set to zero, as depicted in the figure.

It's worth mentioning that Tack (2021) conducted fine-tuning on contextualized BERT models. However, these models were not employed due to their underperformance compared to the previous two models, as indicated by the results presented in Tack (2021).

The two models presented in Figures 1 and 2 were trained using the dataset detailed in Chapter 5 of Tack's thesis, which expanded upon the initial data collected by Tack et al. (2016a). This training dataset comprised 262,054 binary decision annotations gathered from a sample of 56 non-native[2] French readers. These annotations captured *perceived* lexical difficulty, as participants were instructed to read texts and highlight words they did not understand. This method differed from measuring *actual* lexical difficulty. Since participants were prompted to highlight words, they could potentially overlook genuinely challenging words that they didn't recognize while reading the text.

2.2 Test Data for French

The French test data, as supplied by Shardlow et al. (2024a), contained 570 items. Each item included an identifier, the language, contextual word usage, and the target word requiring difficulty prediction, as depicted in Figure 3. Among the total 570 target words, the dataset comprised 560 unique word types and covered 191 distinct sentence contexts. For the lexical complexity prediction track, the French test data was annotated by 10 raters, all of whom were non-native French speakers. Their native languages included Arabic (2), Mandarin (2), German (1), Hindi (1), Italian (1), Japanese (1), Spanish (1), and Turkish (1).

3 Results

Figure 4 illustrates the model predictions for the French test dataset. As shown, both models generally predicted a high difficulty level (> 0.5) for most test items, with the isolated model (run 2) indicating a higher difficulty level compared to the contextualized model (run 1).

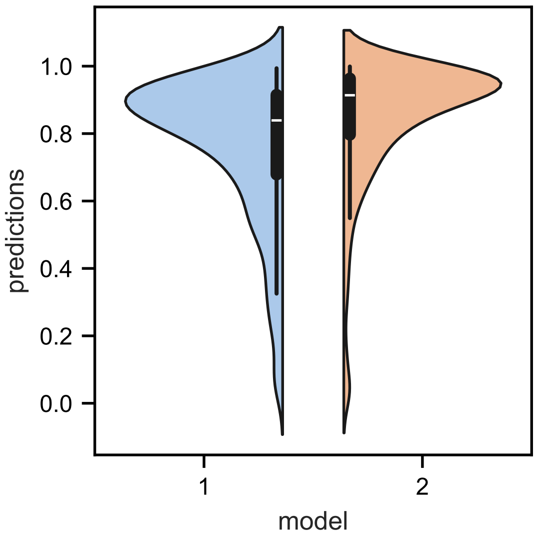


Figure 4: Predictions of Lexical Complexity for the French Test Data

Table 1 showcases the leaderboard results for the French test dataset. Notably, the R^2 metric suggests that both models exhibited a negative fit with the true complexity scores, as supported by the high (worse) scores for MAE and MSE. One likely explanation is that both models were trained

[2] Most readers were native Dutch speakers, with a minority being speakers of Chinese, Japanese, and Spanish.

#	Team	Run	r	ρ	MAE	MSE	R^2
1	TMU-HIT	A	0.6253	0.6302	0.1669	0.0452	0.2704
2	Archaeology	1	0.5335	0.5310	0.1898	0.0487	0.2136
3	TMU-HIT	A	0.5278	0.5343	0.1744	0.0471	0.2391
4	RETUYT-INCO	A	0.4868	0.4651	0.2063	0.0602	0.0279
5	Archaeology	2	0.4411	0.4188	0.1851	0.0504	0.1862
6	Archaeology	A	0.4411	0.4188	0.1851	0.0504	0.1862
→ 7	ITEC	2	0.3607	0.4972	0.5302	0.3373	–4.4459
→ 8	ITEC	1	0.3253	0.3533	0.4545	0.2694	–3.3488
9	GMU	1	0.3193	0.3207	0.2089	0.0589	0.0484
10	GMU	A	0.1557	0.1756	0.2136	0.0617	0.0039
11	SCaLAR	A	0.1035	0.0674	0.2093	0.0616	0.0061

Table 1: Leaderboard of Lexical Complexity Prediction for French Including the Predictions by the Two Models

for a notably distinct prediction task, namely logistic regression instead of linear regression. Another conceivable factor contributing to the negative fit is the variation in native languages among the non-native readers who annotated the data in Tack (2021) compared to those who annotated the French test dataset (see Section 2.2). Since annotators' native languages influence their perception of word difficulty, this variation is likely to impact the accuracy of the predictions.

However, the findings presented in Table 1 also demonstrate that, despite the weak fit, the models still exhibited a modest positive correlation with the true complexity scores. Specifically, the findings indicated that isolated predictions showed a slightly stronger correlation ($r = .36$) compared to contextualized predictions ($r = .33$).

These results align with earlier observations by Tack (2021), indicating that the ground truth predominantly reflects greater challenges in lexical access (i.e., difficulty recognizing the form and meaning of the word, regardless of its context) rather than issues in word-to-context integration (i.e., difficulty in interpreting the word within its context). Specifically, Tack (2021) noted that words identified as challenging by non-native readers exhibited more lexical access difficulties, as indicated by the higher predictive power of features associated with isolated word surprisal compared to contextualized word surprisal. This finding is unsurprising, given that the annotators had elementary to intermediate proficiency levels and, therefore, had a significantly smaller vocabulary size compared to native speakers. Consequently, it is reasonable to assume that the non-native annotators of the French test dataset also had a lower vocabulary size and were thus more susceptible to encountering words not yet ingrained in their mental lexicon, resulting in greater challenges in recognizing the form and meaning of words.

Furthermore, the results depicted in Table 1 reveal that isolated predictions demonstrated a notably higher (and fourth-best) Spearman rank correlation ($\rho = .50$) compared to contextualized predictions ($\rho = .35$). This suggests that although the logistic scores predicted by the model might not closely match the continuous complexity scores, they still preserve the same ranking of difficulty as the continuous complexity scores would. Therefore, even though transferring the difficulty scores may pose uncertainty, there is an interesting potential in transferring the ranking of lexical difficulty from this model to new data.

4 Conclusion

This study delved into predicting lexical complexity in French test data employing two models: an isolated model and a contextualized model. The findings underscore that while the transfer of difficulty scores remains uncertain, the ranking of lexical difficulty from this model can still be applied to new data. This emphasizes the potential usefulness of the models in comprehending lexical complexity in French texts, while also spotlighting the limitations in transferring the raw predicted scores. Moving forward, we also intend to explore the implications of transferring zero-shot predictions made with pre-trained French models to other languages.

References

Dirk De Hertog and Anaïs Tack. 2018. Deep Learning Architecture for Complex Word Identification. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, volume 13, pages 328–334, New Orleans, Louisiana. Association for Computational Linguistics.

Wesley A. Hoover and Philip B. Gough. 1990. The Simple View of Reading. *Reading and Writing*, 2(2):127–160.

Kai North, Tharindu Ranasinghe, Matthew Shardlow, and Marcos Zampieri. 2024. Multils: A multi-task lexical simplification framework. *arXiv preprint arXiv:2402.14972*.

Gustavo Paetzold and Lucia Specia. 2016. SemEval 2016 Task 11: Complex Word Identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569, San Diego, California. Association for Computational Linguistics.

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.

Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Huelsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Peréz Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, Sanja Štajner, Marcos Zampieri, and Horacio Saggion. 2024a. The BEA 2024 Shared Task on the Multilingual Lexical Simplification Pipeline. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*.

Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Huelsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Peréz Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, Marcos Zampieri, and Horacio Saggion. 2024b. An Extensible Massively Multilingual Lexical Simplification Pipeline Dataset using the MultiLS Framework. In *Proceedings of the 3rd Workshop on Tools and Resources for People with REAding DIfficulties (READI)*.

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.

Anaïs Tack. 2021. *Mark My Words! On the Automated Prediction of Lexical Difficulty for Foreign Language Readers*. Ph.D. thesis, UCLouvain & KU Leuven, Louvain-la-Neuve, Belgium.

Anaïs Tack, Thomas François, Anne-Laure Ligozat, and Cédrick Fairon. 2016a. 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*, volume 10, pages 230–236, Portorož, Slovenia. European Language Resources Association.

Anaïs Tack, Thomas François, Anne-Laure Ligozat, and Cédrick Fairon. 2016b. Modèles adaptatifs pour prédire automatiquement la compétence lexicale d'un apprenant de français langue étrangère. In *Actes de la conférence conjointe JEP-TALN-RECITAL 2016. volume 2 : TALN (Articles longs)*, volume 23, pages 221–234, Paris, France. AFCP - ATALA.

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*, volume 13, pages 66–78, New Orleans, Louisiana. Association for Computational Linguistics.

Author Index

Alexandron, Giora, 391
Alfonso, Leandro, 618
Alva-Manchego, Fernando, 571
Alwan, Abeer, 561
Ariely, Moriah, 391
Ashok Kumar, Nischal, 108

Baffour, Perpetual, 99
Baldwin, Peter, 470
Banno, Stefano, 149
Batista-Navarro, Riza, 571
Beigman Klebanov, Beata, 182, 391, 459
Berruti, Santiago, 381
Bexte, Marie, 119
Biermann, Leon, 283
Bott, Stefan, 571
Bouwer, Renske, 125
Bradford, Allison, 450
Brown, Kenneth, 330
Bulut, Okan, 522
Buseyne, Siem, 512

Calderon Ramirez, Saul, 571
Cardon, Rémi, 571
Carpenter, Dan, 403
Carraro, Diego, 330
Chamieh, Imran, 309
Chen, Changsheng, 512
Chen, Xiaobin, 199
Chenna, Suma, 1
Chernodub, Artem, 17
Chiruzzo, Luis, 381, 618
Christ, Oliver, 119
Clauser, Brian, 470
Collazo, Arturo, 381
Correnti, Richard, 365
Cristea, Petru, 610
Crossley, Scott, 99

D'hondt, Robbe, 512
Dainese, Nicola, 165
Dascalu, Mihai, 99
De Vrindt, Michiel, 125, 512
Debaillon, Mathis, 605
Di Mitri, Daniele, 483
Ding, Yuning, 210, 439
Dobnik, Simon, 414
Doi, Kosuke, 316

Drachsler, Hendrik, 483
Duenas, George, 542
Duran Karaoz, Zeynep, 503
Durward, Matthew, 266
Dutilleul, Benjamin, 605

Enomoto, Taisei, 590

Felice, Mariano, 503
Feng, Wanyong, 222
Fleitas, Facundo, 618
François, Thomas, 571
Fulari, Rishikesh, 528

Gales, Mark, 149
Gharahighehi, Alireza, 512
Giebermann, Klaus, 309
Gil, Federico, 618
Glandorf, Dominik, 299
Gombert, Sebastian, 483
Gorgun, Guher, 522
Goswami, Dhiman, 627
Gurin Schleifer, Abigail, 391
Góngora, Santiago, 618

Ha, Le An, 87, 470
Hagiwara, Masato, 251
Harik, Polina, 87, 470
Hasan, Mohammad, 137
Hatami, Zahra, 551
Hayakawa, Akio, 571
Hayat, Ahatsham, 137
Hellas, Arto, 165
Hershcovich, Daniel, 351
Hirasawa, Tosho, 590
Hirschberg, Julia, 34
Horbach, Andrea, 119, 210, 439, 571
Hovy, Dirk, 54
Huelsing, Anna, 571

Ilagan, Michael, 182
Inui, Kentaro, 251
Ionescu, Radu Tudor, 493

Jansen, Thorben, 210, 439
Jimenez, Sergio, 542

Kesanam, Ashinee, 534

Khan, Bilal, 137
Kim, Hwichan, 590
Knill, Kate, 149
Kobayashi, Masamune, 68
Komachi, Mamoru, 68, 590
Korniienko, Oleksandr, 17
Koutcheme, Charles, 165
Kumar, Shankar, 11
Kwako, Alexander, 78

LaFlair, Geoffrey, 428
Lan, Andrew, 108, 222
Lee, John, 343
Lee, Seung, 403
Lester, James, 403
Lesterhuis, Marije, 125
Li, Tianwen, 365
Lim, Ho Hung, 343
Linn, Marcia, 450
Litman, Diane, 365
Liu, Zhexiong, 365
Liubonko, Andrii, 17
Lohmann, Julian, 439
Lucas, Andrés, 618

M, Anand Kumar, 534
Mateus Ferro, Geral, 542
Mathias, Sandeep, 605
Matsumura, Lindsay, 365
Mee, Janet, 87
Menzel, Lukas, 483
Metwaly, Sameh, 512
Meurers, Detmar, 199, 299
Meyer, Jennifer, 210
Mikeska, Jamie, 182
Min, Wookhee, 403
Mita, Masato, 68, 251
Mizumoto, Tomoya, 251
Moreno-Álvarez, Sergio, 567
Mulcaire, Phoebe, 428
Muñoz Sánchez, Ricardo, 414

Nagai, Yoshinari, 590
Nakajima, Kyotaro, 590
Nakamura, Satoshi, 316
Nakano, Felipe Kenji, 512
Nehring, Andreas, 283
Nisioi, Sergiu, 610
Noreillie, Ann-Sophie, 512
North, Kai, 470, 627

O'Reilly, Tenaha, 459
Omelianchuk, Kostiantyn, 17
Ormerod, Christopher, 78
Ozogul, Gamze, 403

Paddags, Benjamin, 351
Peñas, Anselmo, 567

Quranda, Milton, 87

Ray Choudhury, Sagnik, 470
Rezayi, Saed, 470
Ribeiro-Flucht, Luisa, 199
Riordan, BRIAN, 450
Rodrigo, Alvaro, 567
Rogoz, Ana-Cristina, 493
Rooein, Donya, 54
Rosá, Aiala, 381, 618
Runge, Andrew, 428
Rusert, Jonathan, 528
Ruseti, Stefan, 99
Röttger, Paul, 54

Sakaguchi, Keisuke, 251
Samokhin, Igor, 17
Sastre, Ignacio, 618
Sato, Ayako, 590
Savage, Valkyrie, 351
Scaria, Nicy, 1
Scarlatos, Alexander, 222
Schaller, Nils-Jonathan, 210, 439
Schützler, Lena, 119
Sellanes, Diego, 381
Seneviratne, Sandaru, 599
Shaitarova, Anastassia, 54
Shaka, Martha, 330
Shankar, Natarajan Balaji, 561
Shao, Teresa, 34
Shardlow, Matthew, 571
Siyan, Li, 34
Skurzhanskyi, Oleksandr, 17
Smith, Digory, 222
Spoturno, Tomás, 618
Stahl, Maja, 283
Stahlberg, Felix, 11
Steimel, Kenneth, 450
Stowe, Kevin, 232
Subramani, Deepak, 1
Sudoh, Katsuhito, 316
Suen, King Yiu, 87
Suhan, Michael, 459

Suominen, Hanna, 599
Suzuki, Jun, 251

Tack, Anaïs, 125, 512, 635
Tan, Bin, 522
Thapa, Surendrabikram, 561
Thomson, Christopher, 266

Van Den Noortgate, Wim, 125
Veeramani, Hariram, 561
Venkata Ravi Ram, Gummuluri, 534
Volodina, Elena, 414
Vydana, Hari Krishna, 149

Wachsmuth, Henning, 283
Wang, Elaine, 365
Wang, Zuowei, 459
Woodhead, Simon, 222

Yancey, Kevin P., 428
Yaneva, Victoria, 87, 470
Yousefpoori-Naeim, Mehrdad, 551
Yu, Zhou, 34

Zampieri, Marcos, 627
Zargari, Shayan, 551
Zesch, Torsten, 119, 309
Zheng, Xiaoying, 403
Zhou, Yiyun, 470